

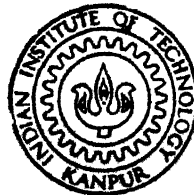
# CRACK PROPAGATION ANALYSIS BY FEM AND BEM : AN INTERACTIVE APPROACH

23.4  
22 m.  
21- 1/9  
20- 2/9

by

RAMENDRA MANDAL

TH  
ME/1988/M  
M32c



DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

DECEMBER, 1988

# **CRACK PROPAGATION ANALYSIS BY FEM AND BEM : AN INTERACTIVE APPROACH**

A Thesis Submitted  
in Partial Fulfilment of the Requirements  
for the Degree of

**MASTER OF TECHNOLOGY**

**118201**

*by*

**RAMENDRA MANDAL**

*to the*

**DEPARTMENT OF MECHANICAL ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR  
DECEMBER, 1988**

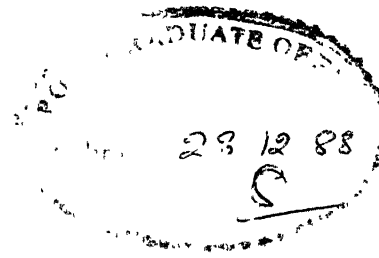
5 OCT 1984

CENTRAL LIBRARY

105914

620-1125

IE-1988-M-MAN-CRA



CERTIFICATE

This is to certify that this work entitled, "CRACK PROPAGATION ANALYSIS USING FEM AND BEM - AN INTERACTIVE APPROACH" by Ramendra Mandal has been carried out under our supervision and has not been submitted elsewhere for a degree.

Dr.S.G. Dhande  
Professor  
Department of Mech.Engg.,  
Indian Institute of  
Technology, Kanpur-208016,  
India

*NN Kishore* 21/12/88  
Dr.N.N. Kishore  
Assistant Professor  
Department of Mechanical Engg.  
Indian Institute of Technology  
Kanpur-208016,  
India

December, 1988.

0568 - 802042  
011



### ACKNOWLEDGEMENTS

I am extremely grateful to Dr. S.G. Dhande and Dr. N.N. Kishore for their inspiring guidance, invaluable suggestions, constructive criticism and being a constant source of encouragement through out this work.

I am thankful to Mr. K.G. Shastry for suggestions during this work. I am also thankful to Research Engineers of CAD-P for extending full cooperation. I sincerely thank all my friends who made my stay here a very pleasant one.

I greatly appreciate the excellent typing of Mr. U.S. Mishra and neat tracing of Mr. A.K. Ganguly.

Finally, I wish to express my gratitude to my parents for their permission and encouragement to pursue graduate study against all odds.

-Ramendra Mandal

## CONTENTS

	<u>Page</u>
LIST OF TABLES	vi)
LIST OF FIGURES	vii)
ABSTRACT	x)
 Chapter 1    INTRODUCTION	 1
1.1    Fracture Mechanics and Discrete Crack Propagation	1
1.2    Interactive Computer Simulation of Crack Propagation	3
1.2.1    Numerical Methods in Fracture Mechanics	4
1.2.2    Mesh Generation, Data Base Design and Interactive Computer Graphics	5
1.3    Review of Previous Works	8
1.3.1    Finite and Boundary Element Crack Propagation Systems	8
1.3.2    Coupling of BEM and FEM	12
1.3.3    Mesh Generation	13
1.3.4    Data Base Design	15
1.4    Objectives of Present Work	16
 Chapter 2    CRACK PROPAGATION CRITERION	 18
2.1    Introduction	18
2.2    Linear Elastic Fracture Mechanics	19
2.3    Strain Energy Density Criterion	20
 Chapter 3    A BRIEF DESCRIPTION OF FEM AND BEM AND THEIR COUPLING	 28
3.1    Introduction	28
3.2    Finite Element Method	28
3.3    Boundary Element Method	34
3.3.1    Boundary Integral Equation	34
3.3.2    Fundamental Solutions	39
3.3.3    Numerical Implementation	41
3.3.4    Stresses and Displacements Inside the Body	48
3.3.5    BEM for Cracked Bodies	50

	<u>Page</u>
3.4 Coupling of BEM and FEM	56
3.4.1 Energy Approach	56
3.4.2 Advantages of Coupling BEM and FEM	61
Chapter 4 INTERACTIVE SIMULATION OF CRACK PROPAGATION	64
4.1 Introduction	64
4.2 Mesh Generation	65
4.2.1 Transfinite Mapping Method	65
4.2.2 Implementation	69
4.3 Database Design	72
4.3.1 Definitions	72
4.3.2 Winged Edge Data Structure Theory	74
4.3.3 Euler Operators	79
4.3.4 Adjacency Relationships	84
4.3.5 Storage and Time Complexity of W-E Data Structure	88
4.3.6 Implementation	90
4.4 Program Integration and A Sample Problem	94
4.4.1 Geometry Handling	98
4.4.2 Attribute Editing	101
4.4.3 Crack Tip Definition	104
4.4.4 Analysis and Remeshing	105
Chapter 5 RESULTS AND DISCUSSIONS	110
5.1 Introduction	110
5.2 Convergence of Symmetric BEM Formulation	111
5.3 A Comparison of Time and Storage Requirements of Coupled BEM-FEM and FEM	114
5.4 Testing of the Program	121
5.5 Propagation of a Centre Crack	126
5.6 Propagation of an Angled Notch	132
Chapter 6 CONCLUSIONS AND RECOMMENDATIONS	141
6.1 Conclusions	141
6.2 Recommendations for Further Study	143
REFERENCES	145

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
4.1	Euler operators	81
4.2	Topological queries	86
5.1	Centre crack problem: Comparison of typical FEM and coupled BEM-FEM solution times	117
5.2	Typical storage requirements for FEM and coupled FEM-BEM mesh	120
5.3	Stress intensity factors for a centre crack: Test problem-1	124
5.4	Stress intensity factors for a bent crack: Test problem -2	124
5.5	Stress intensity factors for a crack emanating from a circular hole: Test problem-3	125
5.6	Values of relevant parameters for various crack steps of growing centre crack	129
5.7	Points of fracture initiation of an angled notch.	136

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
2.1	Crack in an infinite plane	22
2.2	Crack tip core region and strain energy density variation	22
2.3	A typical true stress-true strain diagram in tension for ductile material	23
3.1	Two-dimensional body with domain $\Omega$ and boundary $\Gamma$	37
3.2	General region $\Omega^* + \Gamma^*$ containing body $\Omega + \Gamma$ with the same elastic property	37
3.3	Singular point $\xi$ on the boundary surrounded by part of a circular arc	37
3.4	Different types of quadratic boundary elements: a) Continuous quadratic elements, b) Fully discontinuous quadratic elements	51
3.5	Multidomain discretization of cracked body	51
3.6	Traction singular quarter-point boundary element	55
3.7	Element geometries for SIF computation: a) Mode-I crack, b) Mixed-Mode crack	55

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3.8	Domain divided into finite element and boundary element region	57
4.1	Mapping of polyhedron (in this case a cube) on to the surface of a sphere	76
4.2	The winged-edge data structure	76
4.3	Data structure record organisation: a) Edge record, b) Vertex record, c) Face record	78
4.4	Euler operators	83
4.5	Schemata for edge based data structure: a) Nine relations and three entities, b) Winged edge data structure	87
4.6	Layered organization of data base modification routines	92
4.7	Structure of menu pages in FRAPCO	96
4.8	Crack emanating from a circular hole in a plate subjected to uniaxial tension	97
4.9	Boundary curve description of the example problem.	100
4.10	Mesh generation of the example problem	100
4.11	Mesh editing of the example problem	100
4.12	Attribute editing of the example problem	103
4.13	Attribute editing of the example problem	103
4.14	Crack tip definition of the example problem	103
5.1	Typical BEM mesh for centre slant	112

<u>Figure</u>	<u>Title</u>	<u>Page</u>
5.2	Percentage error in computed SIF for slant crack problem	113
5.3	Centre crack problem: (a) Finite element mesh, (b) Coupled BEM-FEM mesh	115
5.4	Centre crack problem	123
5.5	Branched crack problem	123
5.6	Crack emanating from a hole	127
5.7	Centre crack problem	127
5.8	Critical stress for crack initiation, $\sigma_i$ versus initial half crack length, a	130
5.9	Critical stress for global instability, $\sigma_c$ versus initial half crack length, a	130
5.10	Variation of the difference of critical stress for instability and crack initiation, $\sigma_c - \sigma_i$ , versus initial half crack length, a	131
5.11	Angled notch problem: edge loaded in uniaxial compression	133
5.12	Location of fracture initiation point on surface of ellipse	133
5.13	Angled notch problem: a) Initial mesh, b) Details of mesh around notch	134
5.14	Primary fracture initiation load	138
5.15	Comparison of present analysis path with experimental results and FE analysis.	139

## ABSTRACT

The process of initiation and quasi-static propagation of crack under generalized plane stress and plane strain conditions in an isotropic, linear elastic material is studied. The strain energy density failure criterion is used for predicting crack growth characteristics in which stable crack growth is simulated by predicting a series of crack growth steps corresponding to piecewise loading increments when material ahead of the crack tip absorb a critical amount of elastic strain energy density, a material constant. Unstable crack growth which may lead to global instability, is predicted when the strain energy density factor assumes a critical value, which is again a material constant. The whole crack growth process from initiation to unstable growth is thus characterized by two material constants.

The stress analysis is performed by a combined boundary element method (BEM) and finite element method (FEM) to exploit the advantages of both methods. The coupling of multi-domain BEM and FEM is achieved by minimisation of an energy functional leading to symmetric boundary element 'stiffness' matrix so that it could be easily incorporated into an existing finite element software. Starting with an initial structural discretization of the cracked body, the model automatically computes stress-intensity factors, direction of propagation and amount of propagation for a given loading condition. The



process is repeated, again automatically, until the crack instability occurs or required number of crack steps is reached. 'Traction singular' quarter-point boundary elements which contain the dominant  $1/\sqrt{r}$  traction singularity and dominant  $\sqrt{r}$  displacement variation around the crack tip are used at the crack tips.

The quasi-static crack propagation analysis system requires an 'interactive simulation' environment which makes effective use of recent advances in computers, display devices, automatic mesh generation, computer graphics and data base management systems. The finite element-boundary element mesh generation by transfinite mapping method is implemented in the form of discrete transfinite mappings. The conventional finite element data structure can not provide the program the ability to respond to an analyst's request in a reasonable amount of time. Application of an existing data structure, the winged-edge (W-E), to BEM-FEM coupled analysis system is introduced which can provide the required performance. The W-E data structure and related concepts are described with particular emphasis on its implementation in a combined BEM-FEM fracture mechanics analysis system.

A FRacture Propagation Code (FRAPCO), which incorporates all the above mentioned concepts, i.e. fracture mechanics, BEM, FEM, automatic mesh generation, mesh and attribute editing, data base design, computer graphics etc. is developed and this

integration aspect is described. A number of example problems are solved to show the capability of FRAPCO and to study the validity and desirability of coupling of FEM and BEM for application to interactive crack propagation analysis from accuracy, time and storage complexities point of view.

## CHAPTER 1

### INTRODUCTION

#### 1.1 Fracture Mechanics and Discrete Crack Propagation

At the most basic level the essential feature of fracture phenomenon is the rupture of interatomic bonds in the material. In between the phenomenon involves nucleation, growth and coalescence of micro and macro voids on cracks in the material. At the other end of scale, the process is exhibited in the form of separation of a full-scale structural part due to propagation of the dominant flaw.

Owing to this highly complex nature of the phenomenon, at present there is no consistent single theory dealing satisfactorily with all its relevant aspects. The theories developed in the past to deal with fracture of solids generally treat the subject from one of the three points of view, namely, microscopic or atomic, microstructures and macroscopic or continuum.

From engineering point of view, macroscopic theories of fracture mechanics are used. These are based on the notions of continuum solid mechanics and classical thermodynamics which provide the quantitative working tools for dealing with the fracture of structural materials. It is implicitly assumed that material contains some macroscopic

flaws which may act as fracture nuclei and that the medium is a homogeneous continuum in the sense that the size of the microscopic flaws is large in comparison with the characteristic micro-structural dimension of the material. The problem is then to study the effects of applied load, crack geometry, environmental conditions etc.

Over the years, classical fracture mechanics has grown with the objectivity of predicting the initiation of fracture and addressing the problem of predicting structural failure as the immediate consequence of fracture initiation. But a fracture, once initiated, is not necessarily always unstable. So, for a propagating crack one must try to find out its trajectory, wheather the crack stops after a certain distance and what must be done to get it going again. In other words, fracture propagation can be stable in load control sense. Assuming linear elastic fracture mechanics (LEFM) conditions, this implies that stress-intensity is decreasing with increasing crack length. Obviously, it is not sufficient just to compute stress-intensity factors for an initial crack configuration. One has to update stress-intensity factors as crack length changes.

Propagation of multiple cracks is common in realistic problems of fracture mechanics. One has to accommodate simultaneous propagation of many cracks. Thus with each growth increment of a given crack, a new boundary value problem is generated. Displacement and traction boundary conditions may change, stress fields are altered and loading

may change in direction and intensity. As a consequence, propagation of one crack may cause initiation of another crack and cracks may influence each other's stability and trajectory.

With possibility of so many factors interacting, one must dispense with a simplification often applied to fracture mechanics, that of self-similar propagation. Curvilinear (mixed-mode) crack propagation is common in real life problems of fracture mechanics. Moreover, if one accepts the mixed-mode stress intensity factors, for example,  $K_I$  and  $K_{II}$ , can be present along a crack length, then theoretically, mixed mode fracture initiation can occur even if  $K_I \leq K_{IC}$  where  $K_{IC}$  is the critical stress intensity factor, a material constant. Consequently, one must incorporate an interactive theory which accurately predicts the critical mixture of stress intensity factors which will cause the next increment of propagation.

## 1.2 Interactive Computer Simulation of Crack Propagation

Because of complexity and multitudes of difficulties in modelling crack propagation, it is currently impossible to create a general 'problem in, solution out' or 'expert system' computer code to model fracture propagation. Rather, by increasing interactivity of the code, it is possible to do numerical analysis under continuous, real-time control of the analysis through interactive computer graphics. The necessary components of such an interactive code, namely, numerical method, mesh generation, data base design and

interactive computer graphics are introduced in this section.

#### 1.2.1 Numerical Methods in Fracture Mechanics

In most of the realistic situations, analytical solutions can not be carried out, and numerical solutions has to be restored to. At present, one of the most generally used method having least limitations, is the finite element method (FEM) and another exciting method which has made profound impact on many fields is boundary element method (BEM).

Finite element method can be interpreted as a piecewise application of variational methods in which both domain and boundary of the problem are discretized into number of elements. The governing differential equations of the problem are then approximated over the region by functions which fully or partially satisfy the boundary conditions.

Another possibility is to use approximating functions that satisfy the governing differential equations in the domain exactly but not the boundary conditions. These techniques are called boundary solution method or boundary element methods. Both BEM and FEM will be explained in more detail in the subsequent chapters but it is interesting to note that they can be interpreted as different weighting residual formulations and thus can be coupled in a single unified code to exploit advantages of both.

In both BEM and FEM, the undermined parameters (e.g. displacements) represent the values of the solution at finite number of preselected points called nodes, on the boundary only in case of BEM and on boundary as well as interior of the domain in case of FEM. These values of primary variable can then be used in a post-processing operation to calculate various fracture mechanics parameters such as stress intensity factors, strain energy density, circumferential stresses etc. at desired points on the domain and boundary. These parameters are then used to construe the actual state of the cracked body and also to determine other important parameters such as direction of propagation, amount of propagation etc. to be used in the next iterative step of analysis. Thus a totally interactive simulation of quasi-static discrete crack propagation can be achieved in which user, in conjunction with the program, interacts to control the course of analysis.

#### 1.2.2 Mesh Generation, Data Base Design and Interactive Computer Graphics

Discrete crack propagation analysis makes much more stringent requirements of the FEM and BEM analyses softwares than analyses for which these tools are normally designed. There are three reasons for this:

1. Contemporary numerical fracture mechanics is a task of modelling inherently non-continuous behaviour (cracks and localizations) with inherently continuous numerical techniques (BEM and FEM).

2. Because of (1), the geometric and topologic (mesh) description of the problem must change with each analysis step. Not only must the geometry and topology change but the nature of the change is a function of the previous analysis results. This implies a close coupling between the analyst's front and back-end tools.
3. The geometric and topological complexity of representing an arbitrary fracture surface in an arbitrary body is something which is beyond the capabilities of most commonly used software tools.

Reason one, above, dictates that the analysis results for fracture problems are much more mesh dependent than results of more conventional analyses. In fact, to model these processes objectively one must remesh the problem at each analysis step to reflect the configuration change due to crack propagation. Reason two dictates that there must exist one piece of software which contains a hybrid of traditional pre-and post-processing capabilities. This piece of software extracts the results of one analysis and use it to preprocess a problem for the next analysis. The third reason dictates that to perform numerical fracture mechanics, one should have powerful tools for geometric modelling and automatic finite and boundary element mesh generation. In order to achieve all these capabilities, one must design the program around a data structure that will support a fast query time, dynamic



changes in problem size, easy identification of features of model such as mesh boundaries and material boundaries. Traditional finite and boundary element data structures fail to provide this kind of stringent requirements and some special data structure which has got these capabilities must be restored to.

A necessary component of an interactive simulation environment is computer graphics. A computer graphics subsystem appropriate for simulation of complex fracture events must involve mesh displays, interactive inputting by pointing to some item on the screen, ability to display different pre-selected menu items on screen so as to guide the user during the flow of analysis. In other words, it must provide fast visual feedback to the user to help him perform numerical analysis under continuous, real-time control.

All the concepts discussed so far, namely, fracture mechanics, finite and boundary element analysis, mesh generation, data base design, remeshing etc. combine in a coherent program to produce 'interactive simulation' of crack propagation process. By 'interactive simulation' here is meant numerical analysis under continuous, real time control of the analysis through interactive computer graphics. By 'control' is meant that the pace and the direction of the simulation are dictated by the analyst rather than by a time-sharing operating system and the program algorithms. The philosophy behind the generation of fracture propagation

system to be described here is to provide a fracture mechanics expert with the tools necessary to predict crack growth rate and direction at a given instant in the crack evolution. The user can, using his own expertise, decide to allow the crack to grow further, according to the prediction of the program or to over ride the predictions of the program and decide, for instance, to test his own theory to calculate the crack growth characteristic. The interactivity of the fracture propagation systems provide a flexibility by which the user may gain understanding into the fracture propagation process through numerical experimentation.

### 1.3 Review of Previous Works

#### 1.3.1 Finite and Boundary Element Crack Propagation Systems

A number of fracture propagation programs have been reported in literature. The early version of the programs [1],[2],[3] relied heavily on the available analytical solutions for calculating stress-intensity factors at the crack tip and as such, they were not applicable to arbitrary structures containing cracks under generalised loadings.

Truly generalized fracture propagation systems which can handle any geometry and loading conditions stem out of a number of special crack tip finite elements used for this purpose. First of such type of elements either utilised a series solution as displacement functions [4],[5] or elementary polynomials of higher order [6],[7] as interpolation functions. But these elements have found limited

application in predicting stress intensity factors for un-propagated crack geometry and have not been utilised for propagation study because of their difficulty in satisfying compatibility condition, utilisation of large number of nodes, difficulty in remeshing to reflect crack growth etc.

Another type of element which found some use in propagation study is so called hybrid elements. Hybrid formulation permit choice of two or more assumed fields of behaviour in a single element. In fracture mechanics both assumed stress-hybrid formulation [8] and assumed displacement hybrid formulation [9] have been applied. These special elements could easily be combined with those obtained through usual displacement finite element formulation, keeping inter-element compatibility intact and also, they can be more easily made to handle the crack tip singularity in presence of plastic deformation. The SIFs are usually treated as unknown variables and obtained directly through solution of system equations. Atluri et.al. [10] developed OR-FLAW, a finite element program for such direct evaluation of SIFs for user defined flaws in plates, cylinders and pressure vessel nozzle corners. OR-FLAW is applicable to a rather limited variety of structures because it does not have a generalized preprocessing or mesh-updating capability. Also, hybrid formulation is very complicated and though they have been studied quite extensively, their implementation has been quite limited in general purpose finite element LEFM programs.

Simulation of more realistic crack propagation by FEM have been achieved by Ingraffea et.al. [11] , [12] . In these programs, quarter-point element [62] are used because it gives acceptable accuracy to model crack tip stress and displacement fields, does not require any interpolation scheme to extract SIFs and most importantly, could be easily generated in meshing and remeshing automatically. In [11] , the program FEFAP (Finite Element Fracture Analysis Program) has been described in connection with modelling discrete cracking in reinforced concrete. Though the program has the capability of modelling general crack geometry and multiple cracks along with pre-and post processing attachments ,it has low-level interaction capabilities. Later, a high level interactive version of the same program, FEFAP-G has been reported [12] . Still, it s pre- and post-processing capabilities are limited because it relies on traditional finite element data structure for this purpose.

Recently, in [13] Wawrzynek and Ingraffea described a versatile 2-dimensional finite element program, called FRANC (FRacture ANALysis Code). It utilises quarter-point elements to model crack tip stress-and displacement fields. Very powerful pre-and post-processing facilities have been made available in FRANC. It utilises many new concepts, data structures, algorithms and is the most versatile of all the fracture propagation system described so far.

Boundary element method has also been used for crack surface modelling. Several strategies have been proposed for the analysis of crack problems using BEM. These methods include representing the crack as a notch, symmetric crack modelling, use of special Green's function and flat crack modelling. Representing crack as a notch or replacing crack plane with symmetry boundary conditions removes the singularity in the algebraic system of equations which is obtained when upper and lower crack surfaces are modelled in the same plane [14]. It is, however, limited to symmetric crack problems. Representing the crack as a notch increases the modelling error due to the notch opening. The special Green's function approach [15], [16] has the advantage that crack geometry and crack tip singularities are fully embedded. The disadvantage is that some two-dimensional and all-three dimensional problems cannot be formulated using Green's function approach. Flat-crack modelling [61] represents the displacements along the crack surface as the relative displacement between the two crack surfaces. This scheme has two critical deficiencies as a mathematical model for crack geometries, as pointed out by Cruse [61]. First, if there is no traction on the boundary and only crack surface loading, a non-unique boundary integral equation is generated. Second, two unknown displacement variables, i.e., the relative and total displacements, exist along the crack. A possible solution for these problems is given by Cruse [61].

An alternative strategy for modelling the crack and subsequent SIF computations is given by Blandford et.al. [18]. In this, multidomain discretization is used to model two crack surfaces in two separate domains which eliminates the problems mentioned above. Perucchio et.al. [19] extended this multidomain boundary element crack modelling technique to three-dimensional cases.

Later, boundary element method is used in crack propagation study. Ingraffea et.al. [20] has used boundary element to model discrete crack propagation in two dimensions. It utilises multi-domain BEM and as such, can handle arbitrary geometry and crack shape. Gerstle et.al. [12] described a three dimensional boundary element program BEM-3D which models three-dimensional crack geometry by multidomain BEM. It has got extensive graphics attachment for doing various pre- and post-processing jobs. But none of the boundary element programs described here uses any special data structure support and as such they are slow in responding to various user's request such as mesh editing and remeshing.

### 1.3.2 Coupling of BEM and FEM

In order to exploit the advantages of both BEM and FEM, the validity and desirability of coupling of these two methods have been studied extensively. Such coupling of BEM and FEM for static problems was first proposed by Zienkiewicz [21] and Shaw [22]. In these papers, integral equation was used in connection with an energy functional. The method gives rise to

symmetric 'stiffness' matrices of boundary element region thus making it ideal for incorporating into an existing finite element software.

Brebbia and Gergiou [23] presented a direct method of combining these two methods and they compared symmetrized direct stiffness matrix with conventional BEM. Kelly et.al. [24] compared the non-symmetric and symmetric stiffness formulations for potential problems. A critical study of various methods of generating boundary element 'stiffness' matrices for solid mechanics application can be found in [25], [26], [27]. Kishimoto et.al. [17] have applied a coupled BEM-FEM method for elastic-plastic fracture mechanics. A few finite elements have been used around the crack tip plastic zone and boundary elements over rest of the domain, as FEM can handle plasticity more easily. Stress-intensity factors have been extracted and they correspond quite well with previously published results. But Kishimoto et.al. [17] restricted such application to an initial crack geometry and crack propagation study by a coupled BEM-FEM has not been attempted.

### 1.3.3 Mesh Generation

Current mesh generators employ one or a combination of two of the following techniques:

- (1) Node connection approach
- (2) Grid-based approach
- (3) Smoothing procedures
- (4) Blending functions.

In node connection approach, nodes are first generated inside the domain by some random node generation techniques [28], and the elements are then formed by connecting them to form triangles [29] or quadrilateral [30].

The grid-based approach arises out of the observation that grid looks like a mesh and it can be made into a mesh provided that the grid cells along the object boundary can be turned into elements. Thacker et.al. [31] are probably the first to publish such a mesh generator. Object can also be represented as a CSG [32]. Yerry and Shephard [33] followed a quad-tree encoding.

Smoothing procedures [34], [35] too, are cast in the form of mesh generators. However, they are normally used in conjunction with one of the other techniques to improve the shape of the elements by repositioning the nodal location of mesh generated. They are basically base on Laplacian mesh generator, which is attributed to Wilson [36].

The blending function or mapped element approach requires an object be subdivided manually into simple regions, each of which consists of three or four sides. A mesh can be induced in it by mapping a mesh template of unit triangle or square in the parametric space to the region [37], [38]. This blending technique can be formulated in terms of the elegant projector theory [39] to generalize the concept.

More extensive bibliography on mesh generation techniques can be found in [40], [67].



#### 1.3.4 Data Base Design

The interactive finite and boundary element programs which are to support real time crack propagation analysis must have a data structure which should provide -- a fast query time, dynamic change in problem size, easy identification features of the model such as mesh boundaries, transformation of data into a form that is more efficient for the computationally intensive equation-solving portion of the analysis. An existing data structure, the winged-edge (W-E), originally designed for a very different purpose can be used to provide the above mentioned features. W-E data structure was introduced by Baumgart [41] to describe the surface topology of polyhedra for computer vision. This data structure was later used in boundary representation type of solid modeller [42]. Different variations of the original form of W-E data structure also exist [43], [44].

Weiller [44] has established the validity of such a data structure for use in boundary representation type solid modeller from topological and graph theoretic point of view. Later a combinatorial analysis of the W-E and other such data structures has been presented by Woo [45]. Mantyla [46] has shown how such type of data structures could be used in a solid modelling environment in conjunction with the so called Euler operators for extensive and fast geometric modelling.

Application of such a data structure in finite element is very new and novel. Wawrzynek et.al. [47] describes such

application for two dimensional finite element application, in which a modified winged-edge data structure has been established as a robust data structure which can provide the stringent performance requirements of an interactive fracture propagation system.

#### 1.4 Objectives of Present Work

Development of a comprehensive fracture propagation analysis system utilising FEM and BEM is attempted here. The specific objectives of this work can be listed as:

- i) To study of desirability and validity of applying a coupled BEM-FEM analysis to interactive fracture propagation analysis so as to exploit the advantages of both the methods.
- ii) To develop a computer code utilising a coupling strategy which permits assembling of contribution from boundary element region with adjoining finite elements so that existing finite element software can be utilised with minor or no modifications.
- iii) To make both BEM and FEM or their coupled analysis formulation available in a single code so that user can use any of them at his own discretion thus making the debate regarding which method is better for a particular problem superfluous.

- iv) To utilise existing data structure and mesh generation techniques so that a unified approach could be adopted for interactive generation and modification of both boundary and finite element mesh and problem attributes in real-time environment.
- v) To study storage and time complexity requirements of a coupled BEM-FEM analysis system that is built around an edge based data structure and compare it with finite element method.
- vi) To integrate the concepts of fracture mechanics, finite and boundary element analysis, mesh generation ,data base design, remeshing etc. into a single coherent computer code to provide interactive fracture propagation analysis tools at user's disposal so that he can do numerical experimentation with any of his own theory or concept for accepting or rejecting it.

## CHAPTER 2

### CRACK PROPAGATION CRITERION

#### 2.1 Introduction

The determination of fracture initiation from an existing flaw requires knowledge of stress intensity around the crack tip, determined analytically or numerically as functions of geometry and load, and the appropriate fracture toughness, a material state property, determined experimentally. These parameters combine in a theoretical Mixed-mode fracture initiation function analogous to a multi-axial stress state yield functions of plasticity. A number of theories such as energy release rate, maximum circumferential stress etc. are proposed, though the need of a criterion that can predict failure in a simple and unified manner has not been fulfilled [48].

Recently, Sih [49] , [50] has proposed a theory of fracture based on the field strength of the local strain energy density in an element of material ahead of the crack or notch tip. The most important feature of this criterion is that it is free from physical constrictions and sufficiently general for treating engineering problems encountered in practice. In this chapter, the basic concepts of linear elastic fracture mechanics are described in

section 2.2 and then strain energy density criterion for incremental crack growth is outlined in section 2.3.

## 2.2 Linear Elastic Fracture Mechanics

Linear elastic fracture mechanics analysis neglects the inelasticity at the crack tip such as microcracking and plasticity, which are considered to be restricted as to a sufficiently small volume such that they can be neglected.

Considering a crack problem in an infinite domain, Williams [51] has shown that the stresses for the traction free crack may be written in terms of an infinite series with respect to the polar coordinates  $r$  and  $\theta$ . Substitution of Irwin's [52] definition of stress intensity factors into Williams eigenfunction expressions yields (Fig. 2.1),

$$\begin{aligned}
 \sigma_{11} &= \frac{K_I}{\sqrt{2\pi r}} \cos \theta/2 (1 - \sin \theta/2 \sin 3\theta/2) \\
 &\quad - \frac{K_{II}}{\sqrt{2\pi r}} \sin \theta/2 (2 + \cos \theta/2 \cos \theta/2) + \dots \\
 \sigma_{22} &= \frac{K_I}{\sqrt{2\pi r}} \cos \theta/2 (1 + \sin \theta/2 \sin 3\theta/2) + \frac{K_{II}}{\sqrt{2\pi r}} \\
 &\quad \sin \theta/2 \cos \theta/2 \cos 3\theta/2 + \dots
 \end{aligned} \tag{2.1}$$

$$\sigma_{12} = \frac{K_I}{\sqrt{2\pi r}} \cos \theta/2 \sin \theta/2 \cos 3\theta/2 + \frac{K_{II}}{\sqrt{2\pi r}}$$

in which  $K_I$  and  $K_{II}$  are Mode I and Mode II stress intensity factors (SIF) respectively .

Integration of eqn. (2.1) using strain-displacement and stress-strain relations yields the following relationship for displacements relative to crack tip for  $r \ll a$  ( $a$  = crack length) [53] ,

$$u_1 = \frac{K_I}{4G} \sqrt{\frac{r}{2\pi}} [(2k-1) \cos \theta/2 - \cos 3\theta/2] + \frac{K_{II}}{4G} \sqrt{\frac{r}{2\pi}} [(2k+3) \sin \theta/2 + \sin 3\theta/2] \quad (2.2)$$

$$u_2 = \frac{K_I}{4G} \sqrt{\frac{r}{2\pi}} [(2k-1) \sin \theta/2 - \sin 3\theta/2] - \frac{K_{II}}{4G} \sqrt{\frac{r}{2\pi}} [(2k-3) \cos \theta/2 + \cos 3\theta/2]$$

in which  $G$  = shear modulus,  $k = (3 - 4\nu)$  for plane strain and  $k = (3 - \nu)/(1 + \nu)$  for plane strain,  $\nu$  = Poisson's ratio.

### 2.3 Strain Energy Density Criterion

Strain energy density criterion makes use of strain energy density function  $dW/dV$  which in general can be determined at any point from the relation [48] ,

$$\frac{dW}{dV} = \int_0^{\epsilon_{ij}} \sigma_{ij} d\epsilon_{ij} \quad (2.3)$$

where  $\sigma_{ij}$  and  $\epsilon_{ij}$  are the stress and strain components respectively. In the linear elastic range thus one gets  $dW/dV = \sigma_{ij} \epsilon_{ij}/2$ .

In strain energy density criterion, the first attention is focussed on material elements at a finite distance  $r_0$  from the point of failure initiation e.g. the crack tip (Fig. 2.2). The circular region of radius  $r_0$  represents the core region in which the continuum model fails to describe in detail the state of stress and deformation. The values of strain energy density  $dW/dV$  are calculated along the boundary of the core region and attention is concentrated on the stationary values of  $dW/dV$ . Thus it is assumed that the direction of the element that initiates fracture or yielding corresponds to the minimum or maximum value of the strain energy density function,  $(dW/dV)_{\min}$  or  $(dW/dV)_{\max}$ .

The critical values of  $dW/dV$  for yielding and fracture are obviously different. Referring to the true stress-true strain curve of the material (Fig. 2.3), the critical value of  $dW/dV$  for yielding,  $(dW/dV)_{\max}^C$ , is equal to the area under this diagram upto the point of yielding, while the critical value of  $dW/dV$  for fracture,  $(dW/dV)_{\min}^C$  is equal to the area of the diagram upto the point of fracture. For brittle materials, the values of  $(dW/dV)_{\max}^C$  and  $(dW/dV)_{\min}^C$  are almost equal, while for ductile materials  $(dW/dV)_{\min}^C$  is always greater than  $(dW/dV)_{\max}^C$ . This fact combined with the previous observation that yielding and fracture are associated with

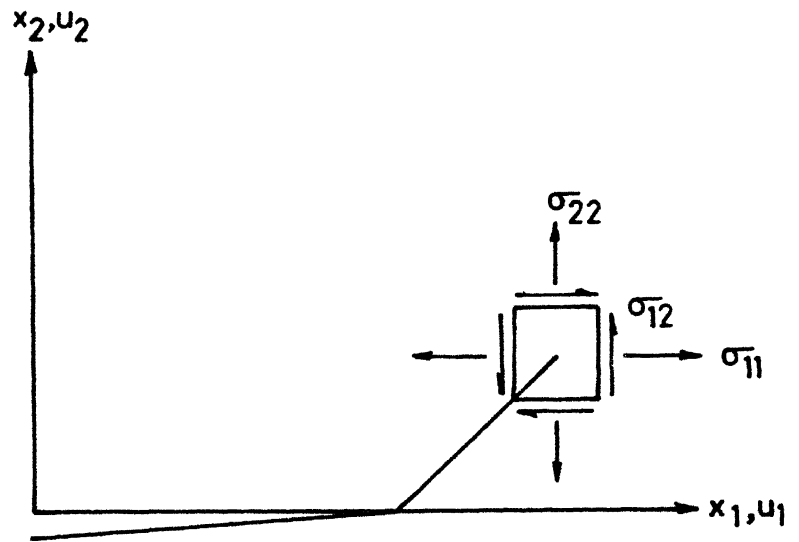


FIG. 2.1 Crack in an infinite plane

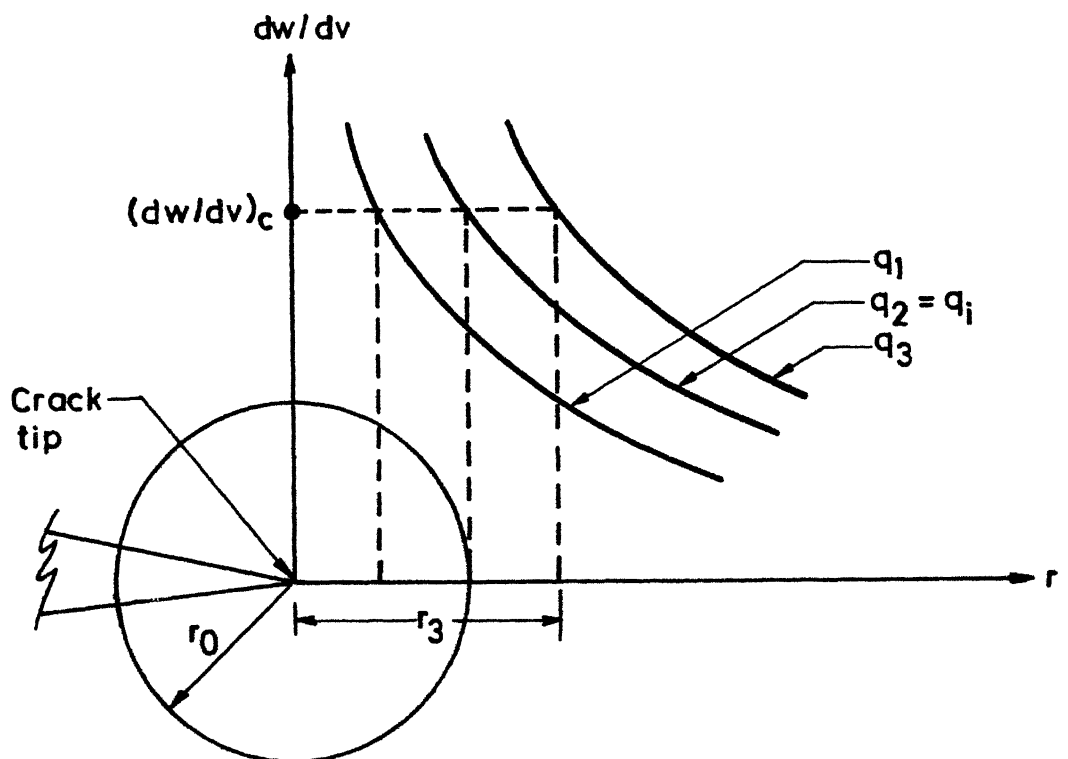


FIG. 2.2 Crack tip core region and strain energy density variation.



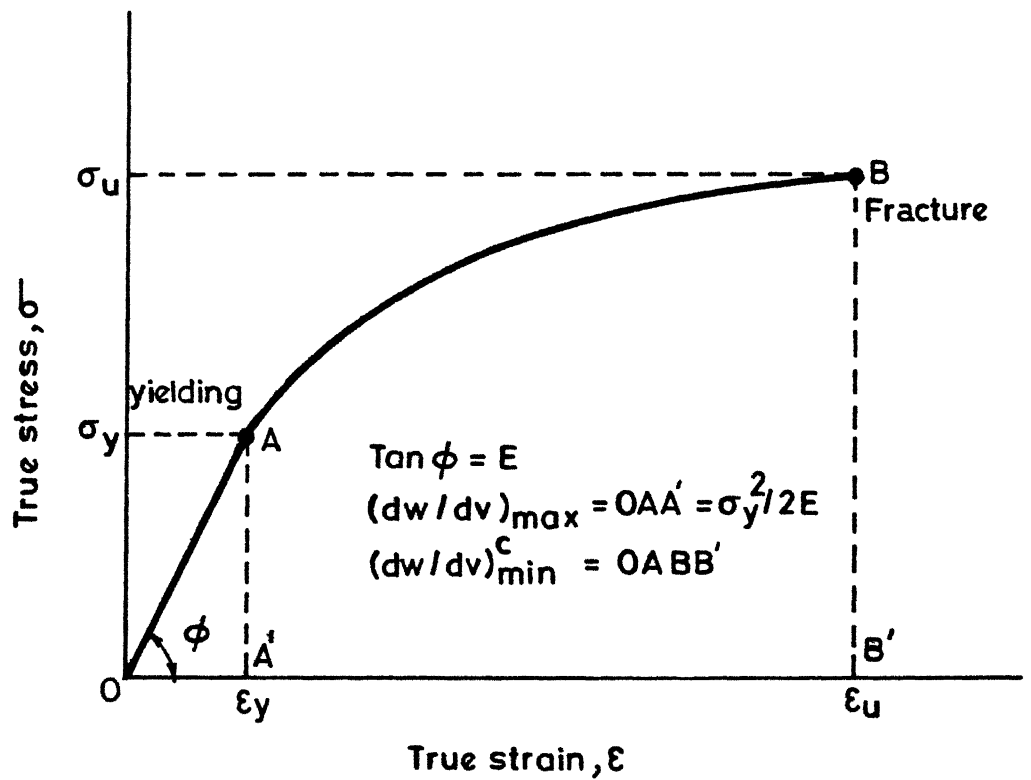


FIG. 2.3 A typical true stress – true strain diagram in tension for ductile material

the maximum and minimum value of  $dW/dV$  respectively, suggests that in ductile materials yielding always preceeds fracture initiation.

Based on the above arguments, the basic hypotheses of the strain energy density criterion for fracture and/or yielding may be stated as [48] ,

Hypothesis (1): The location of fracture coincides with the location of minimum strain energy density  $(dW/dV)_{\min}$ , and yielding with the maximum strain energy density  $(dW/dV)_{\max}$ .

Hypothesis (2): Failure by stable fracture on yielding occurs when  $(dW/dV)_{\min}$  or  $(dW/dV)_{\max}$  reach their respective critical values.

Hypothesis (3): The amount of incremental growth  $r_1, r_2, \dots, r_j, \dots, r_c$  is governed by

$$\left(\frac{dW}{dV}\right)_c = \frac{S_1}{r_1} = \frac{S_2}{r_2} = \dots = \frac{S_j}{r_j} = \dots = \frac{S_c}{r_c} = \text{constant}, \quad (2.4)$$

where the quantity  $S$  is defined by,

$$S = \left(\frac{dW}{dV}\right) r \quad (2.5)$$

There is unstable fracture or yielding when the critical ligament size  $r_c$ , a material constant, is reached. The quantity  $S$  represents the local energy release for a segment of crack growth  $r$ . It is to be noted that eqn.(2.5) is independent of constitutive relations of the material.

Considering the variation of strain energy density function  $dW/dV$  versus distance  $r$  along the direction of expected crack growth (Fig. 2.2), one observes that crack initiation starts for the load for which the  $dW/dV - r$  curve passes from the point defined by the radius  $r_0$  of the core region and  $dW/dV = (dW/dV)_{\min}^c = (dW/dV)_{cr}$ . Thus, in Fig.(2.2), the crack starts to propagate for  $q_{cr} = q_2$  while for  $q = q_1$  the crack does not grow and for  $q = q_3$ , it grows by an amount  $r = r_3$ . Unstable crack growth leading to global instability takes place when  $r = r_c$ , where critical distance  $r_c$  is defined from relation (2.4). This is equivalent to  $S = S_c$  where  $S_c$  represents the fracture toughness of the material.

As pointed out earlier, when inelastic deformation is negligibly small everywhere in solid, linear elastic fracture mechanics can be used to describe the state of stress and strain. Thus the quantity  $dW/dV$  can be written as [49] ,

$$\frac{dW}{dV} = \frac{1}{4G} \left[ \frac{k+1}{4} (\sigma_{11} + \sigma_{22})^2 - 2 (\sigma_{11} \sigma_{22} - \sigma_{12}^2) \right] \quad (2.6)$$

Introducing eqn. (2.1) into eqn. (2.6), the following quadratic form for the strain energy density is obtained,

$$\frac{dW}{dV} = \frac{1}{r} (a_{11} K_I^2 + 2a_{12} K_I K_{II} + a_{22} K_{II}^2) \quad (2.7)$$

where the coefficients  $a_{ij}$  ( $i, j = 1, 2$ ) are given by,

$$16G a_{11} = (1 + \cos \theta) (k - \cos \theta) \quad (2.8a)$$

$$16G a_{12} = \sin \theta [2 \cos \theta - (k-1)] \quad (2.8b)$$

$$16G a_{22} = (k+1) (1 - \cos \theta) + (1+\cos \theta) (3\cos \theta - 1) \quad (2.8c)$$

Substituting eqn. (2.7) into eqn. (2.5), one obtains expression for strain energy density factor  $S$ ,

$$S = a_{11} K_I^2 + 2a_{12} K_I K_{II} + a_{22} K_{II}^2 \quad (2.9)$$

Thus one can now express hypothesis (1) of strain energy density criterion in mathematical form,

$$\frac{\partial S}{\partial \theta} = 0, \quad \frac{\partial^2 S}{\partial \theta^2} > 0 \quad (2.10)$$

On substitution of eqn. (2.9) into eqn. (2.10), the final form is obtained,

$$[2 \cos \theta - (k-1)] \sin \theta K_I^2 + 2[2 \cos 2\theta - (k-1)$$

$$\cos \theta] K_I K_{II} + (k-1 - 6 \cos \theta) \sin \theta K_{II}^2 = 0 \quad (2.11a)$$

$$[2 \cos 2\theta - (k-1) \cos \theta] K_I^2 + 2[(k-1) \sin \theta - 4 \sin 2\theta] K_I K_{II} \\ + [(k-1) \cos \theta - 6 \cos 2\theta] K_{II}^2 > 0 \quad (2.11b)$$

Relations (2.11a) and (2.11b) represents the general formulas of the strain energy density criterion and they are applied in conjunction with hypothesis (2) and (3) and eqn. (2.4), to predict crack growth characteristics. Algorithmically, at any load step, values of stress intensity factors are extracted from stress analysis of the body under consideration and they are introduced into eqn. (2.11) to obtain angle  $\theta$ , which predicts the direction of propagation with respect to local crack axis; the crack starts to grow only if strain energy density  $dW/dV$ , given by eqn. (2.7), along the direction of crack extension at a distance  $r_0$  from the crack tip reaches the critical level  $(dW/dV)_c$  and grows by an amount  $r$ , given by eqn. (2.4).

## CHAPTER 3

### A BRIEF DESCRIPTION OF FEM AND BEM AND THEIR COUPLING

#### 3.1 Introduction

This chapter briefly outlines the theory of finite element and boundary element methods in regard to the stress analysis problems of fracture mechanics. Since finite element method is now well standardized, only a brief discussion is given in section 3.2, while boundary element method has been described in section 3.3 in greater detail, because of its recent origin. The chapter is concluded with a note on the method employed to couple BEM and FEM and advantages gained by such coupling.

#### 3.2 Finite Element Method

In this section, the finite element formulation based on weighted residual technique is presented, basically, with reference to elastostatics problems. Even though the problem could be tackled in many different ways, the weighted residual approach seems to be particularly attractive because of its generality and also, because boundary element and finite element procedures could be treated as essentially two different types of formulation from same family of weighted residual technique, thus facilitating their coupling.

### 3.2.1 The Weighted Residual Process

Consider an elastic body defined by domain  $\Omega$  and boundary  $\Gamma$  (Fig. 3.1). Its equilibrium is established by analysing a system of forces and moments for a generic point. Thus, equilibrium equation is obtained,

$$\sigma_{ij,i} + b_j = 0 \quad (3.1)$$

where  $\sigma_{ij}$  is the stress tensor and  $b_j$  are the body forces. The traction and displacement boundary conditions are given as,

$$\begin{aligned} p_i &= \sigma_{ji} n_j = \bar{p}_i \quad \text{on} \quad \Gamma_2 \\ u &= \bar{u}_i \quad \text{on} \quad \Gamma_1 \end{aligned} \quad (3.2)$$

where  $n_j$  is the outward normal.

The general weighted residual statement can be written as [54]

$$\int_{\Omega} (\sigma_{jk,j} + b_k) u_k^* d\Omega + \int_{\Gamma_2} (p_k - \bar{p}_k) \bar{u}_k^* d\Gamma + \int_{\Gamma_1} (u_k - \bar{u}_k) p_k^* d\Gamma \quad (3.3)$$

where  $u_k^*$  and  $p_k^*$  are the displacements and tractions corresponding to the weighting field, that is,  $p_k^* = n_j \sigma_{jk}^*$ . In FEM, the displacement boundary conditions are satisfied exactly, thus making the residual on  $\Gamma_1$  portion of the boundary zero. So the eqn. (3.3) can be written as,

$$\int_{\Omega} (\sigma_{jk,j} + b_k) u_k^* d\Omega + \int_{\Gamma_2} (p_k - \bar{p}_k) \bar{u}_k^* d\Gamma = 0 \quad (3.4)$$

Integrating the first term in eqn. (3.4) by parts, the weak statement is obtained,

$$\begin{aligned} - \int_{\Omega} \sigma_{jk} \epsilon_{jk}^* d\Omega + \int_{\Omega} b_k u_k^* d\Omega + \int_{\Gamma_2} (p_k - \bar{p}_k) \bar{u}_k^* d\Gamma \\ + \int_{\Gamma_1 + \Gamma_2} p_k u_k^* d\Gamma = 0 \end{aligned} \quad (3.5)$$

Limiting now the choice of weighting functions, so that,

$$\begin{aligned} u_k^* &= 0 \text{ on } \Gamma_1 \\ \bar{u}_k^* &= -u_k^* \text{ on } \Gamma_2 \end{aligned} \quad (3.6)$$

and putting eqn. (3.6) into eqn. (3.5), the final form of weak statement is obtained,

$$\int_{\Omega} \sigma_{jk} \epsilon_{jk}^* d\Omega = \int_{\Omega} b_k u_k^* d\Omega + \int_{\Gamma_2} \bar{p}_k u_k^* d\Gamma \quad (3.7)$$



The independent variable  $u$  is approximated by a finite series (writing in matrix form),

$$\{u\} = \sum_{m=1}^M [N_m] \{\bar{u}_m\} \quad (3.8)$$

where  $[N_m]$  are the shape, trial or basis functions,  $\{\bar{u}_m\}$  is a vectors whose components are the approximations to the displacements at node  $m$ , and  $M$  is the total number of nodes

The strains at any point are obtained, by applying an operator  $L$  on  $\{u\}$ ,

$$\{\epsilon\} = L \sum_{m=1}^M [N_m] \{\bar{u}_m\} = \sum_{m=1}^M [B_m] \{\bar{u}_m\} \quad (3.9)$$

and consequently the stresses are,

$$\{\sigma\} = [D] \{\epsilon\} = [D] \sum_{m=1}^M [B_m] \{\bar{u}_m\} \quad (3.10)$$

where  $[B]$  is the strain-displacement matrix, and  $[D]$  is the stress-strain matrix. The expressions for  $L$ ,  $[B]$ ,  $[D]$  can be found in [54]. Now using the shape functions as the weighting functions (Galerkin method), and substituting for stresses and strains, in terms of displacements from eqn. (3.9) and eqn. (3.10), eqn. (3.7) can be written as, in matrix form (dropping the body force term),

$$\sum_{m=1}^M \left[ \int_{\Omega} [B_1]^T [D] [B_m] d\Omega \right] \{\bar{u}_m\} = \int_{\Gamma_2} [N_1] \{\bar{p}\} d\Gamma \quad (3.11)$$

which is a set of algebraic equations of the form,

$$[K] \{u\} = \{f\} \quad (3.12)$$

where, the matrices  $[K]$  and  $\{f\}$  are obtained by summing the contribution from the individual element matrices, whose components are defined by

$$[K_{lm}^e] = \int_{\Omega^e} [B_1^e][D][B_m^e] d\Omega \quad (3.13)$$

$$\{f_1\} = \int_{\Gamma_2^e} [N_1^e]\{\bar{P}\} d\Gamma$$

The presentation so far has been completely general and, as such, can be applied to any element form, from a wide variety of elements now available. For two dimensional stress analysis, 8-noded isoparametric element is very useful because it can be used to model geometry with curved boundaries and it gives satisfactory degree of accuracy in the solution. This element, details of which can be found in [54] is adopted in the present application.

The expression (3.12) represents the overall equilibrium equations, where matrix  $[K]$  is the overall stiffness matrix or global stiffness matrix and  $\{f\}$  is the global load vector. The matrix  $[K]$  is a singular matrix as given in eqn.(3.12) and can be solved only after incorporating the specified displacement boundary conditions.

The solution of eqn. (3.12) constitute a major part of the required computer time and has a great bearing on the cost of analysis. This has provided a great motivation for the development of various techniques. One particular technique called frontal solution method [55] , which introduces a natural substructuring or partitioning to make the maximum use of uncoupling effect between the nodal degrees of freedom, has become very popular. Requirement of very small core storage and possibility of natural node numbering of added degrees of freedom (such as due to propagation of crack) without any node renumbering scheme, makes it an attractive choice for the present application. Particularly, its ability to solve for many load conditions without recomputing and reassembling the global stiffness matrix is very useful in crack extension study, where, several load increments may be required to drive a crack to the onset of propagation.

The solution of system of eqn. (3.12) gives nodal displacements. The element stresses are then calculated at the element Gauss-points from eqn. (3.10).

For analysis of bodies containing crack, the standard elements such as an eight-noded isoparametric element, fail to give accurate result as they do not contain the dominant  $\sqrt{r}$  term in their displacement interpolation function. Such  $\sqrt{r}$  variation of displacement and  $1/\sqrt{r}$  stress variation around the crack tip can easily be achieved using standard eight-noded isoparametric elements

with 'mid-side' nodes displaced from their original position to the quarter point locations [ 62 ].

### 3.3 Boundary Element Method

Boundary element method, as the name suggests, needs discretization of the boundary of the domain only, because it chooses the trial function in such a manner that the differential equations are satisfied through out the domain a priori. BEM also falls under the weighted residual scheme in broader sense, though, as will be seen in this section, special techniques are applied in its formulation which make it radically different from finite element method.

#### 3.3.1 Boundary Integral Equation

Here again, the general weighted residual statement (eqn. 3.3) is integrated by parts and the constitutive relations, i.e.  $\sigma_{ij} = C_{ijkl} \epsilon_{kl}$  and  $\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i})$  are substituted to give;

$$\begin{aligned} & - \int_{\Omega} C_{jkl i} \epsilon_{li} \epsilon_{jk}^* d\Omega + \int_{\Omega} b_k u_k^* d\Omega \\ & = - \int_{\Gamma_2} \bar{p}_k u_k^* d\Gamma - \int_{\Gamma_1} p_k u_k^* d\Gamma + \int_{\Gamma_1} (\bar{u}_k - u_k) p_k^* d\Gamma \end{aligned}$$

(3.14)

Integrating by parts again the first term in eq. (3.14) and using the reciprocity principle due to symmetry of  $C_{ijkl}$ ,

$$\int_{\Omega} \sigma_{jk} \epsilon_{jk}^* d\Omega = \int_{\Omega} \epsilon_{jk} \sigma_{jk}^* d\Omega \quad (3.15)$$

One obtains

$$\begin{aligned} \int_{\Omega} (\sigma_{jk,j}^*) u_k d\Omega + \int_{\Omega} b_k u_k^* d\Omega = & - \int_{\Gamma_2} \bar{p}_k u_k^* d\Gamma - \int_{\Gamma_1} p_k u_k^* d\Gamma \\ & + \int_{\Gamma_1} \bar{u}_k p_k^* d\Gamma + \int_{\Gamma_2} u_k p_k^* d\Gamma \end{aligned} \quad (3.16)$$

Taking into consideration that the body forces are unknown functions, the second integral on the left hand side of eqn. (3.16) does not introduce any unknowns in the domain  $\Omega$ . The first integral, however, presents unknown displacements in the  $\Omega$  domain. The objective of boundary element method is to eliminate the integral in the domain - the first integral on the left hand side - by proposing weighting functions which satisfy the equilibrium equation in  $\Omega$ . In other words, one looks for Green's function or fundamental solutions satisfying,

$$\sigma_{jk,j}^* + \Delta(\xi, x) e_k = 0 \quad (3.17)$$

in which  $\Delta(\xi, x)$  represents the Dirac Delta function,  $\xi$  is the singular load-point, and,  $x$  is the field point. Here  $x$  and  $\xi$  belong to a generalized domain  $\Omega^*$ , which encompasses the domain of interest  $\Omega$  (Fig. 3.2). This effectively means that  $b_k^*$  is assumed to correspond to a positive unit load applied at point  $\xi \in \Omega^*$  in each of the three orthogonal directions given by the unit vectors  $e_i$ , i.e.

$$b_k^* = \Delta(\xi, x) e_k \quad (3.18)$$

Furthermore, as the starred displacements and tractions,  $u_k^*$  and  $p_k^*$ , represent fundamental solutions, so they can be written as

$$u_k^* = u_{ik}^* (\xi, x) e_i, \quad (3.19)$$

$$p_k^* = p_{ik}^* (\xi, x) e_i.$$

where  $u_{ik}^* (\xi, x)$  and  $p_{ik}^* (\xi, x)$  represent the displacements and tractions in the  $k$  direction at point  $x$  corresponding to a unit point load acting in  $i$  direction ( $e_i$ ) applied at point  $\xi$ .

Substituting eqn. (3.17) and (3.19) into eqn. (3.16) and noting the property of Dirac delta function, i.e.,

$$\int_{\Omega} \Delta(\xi, x) u_i (x) e_i d\Omega = u_i (\xi) e_i \quad (3.20)$$

one obtains, in general,

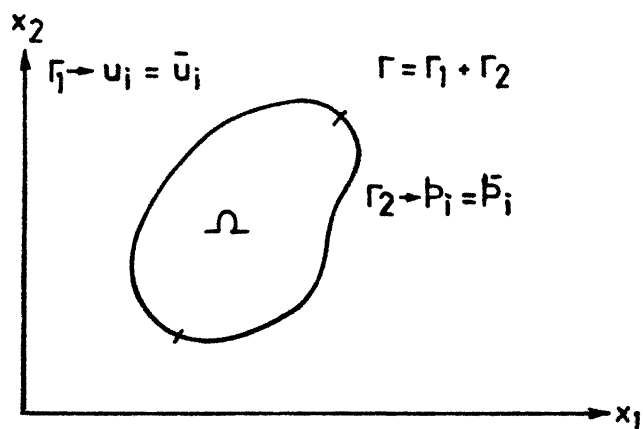


FIG. 3.1 Two-dimensional body with domain  $\Omega$  and boundary  $\Gamma$

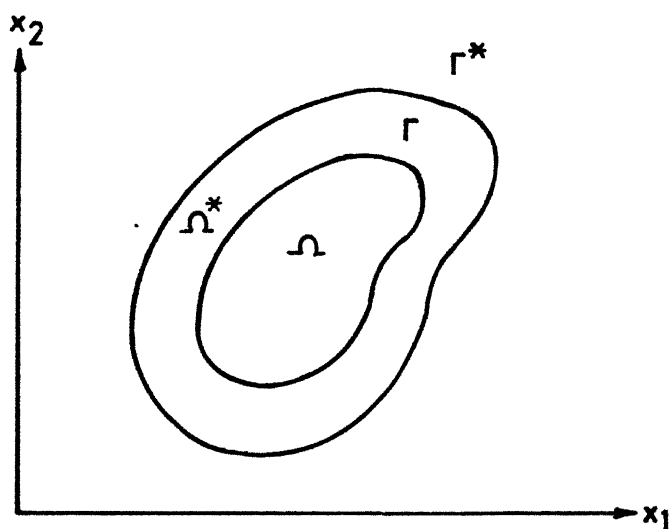


FIG. 3.2 General region  $\Omega^* + \Gamma^*$  containing body  $\Omega + \Gamma$  with the same elastic property

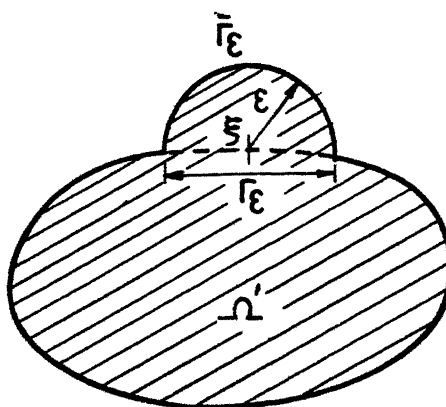


FIG. 3.3 Singular point  $\xi$  on the boundary surrounded by part of a circular arc.

$$\begin{aligned}
u_i(\xi) = & \int_{\Gamma} u_{ij}^*(\xi, x) p_j(x) d\Gamma(x) - \int_{\Gamma} p_{ij}^*(\xi, x) u_j(x) d\Gamma(x) \\
& + \int_{\Omega} u_{ij}^*(\xi, x) b_j(x) d\Omega(x) \quad (3.21)
\end{aligned}$$

Equation (3.21), obtained by weighted residual technique, is known as Somigliana's identity [56] for displacement and can also be obtained using extended version of Betti's work theorem.

Somigliana's identity is a continuous representative of displacements at any point  $\xi \in \Omega$  for known boundary displacements and tractions. The extension of it to a boundary point  $\xi \in \Gamma$  is obtained by increasing the domain around the point under consideration (Fig. 3.3) by a circular region of positive radius and then applying eqn. (3.21) and taking limit when  $\varepsilon \rightarrow 0$ . This gives the final expression of boundary integral equation [57],

$$\begin{aligned}
C_{ij}(\xi) u_j(\xi) + \int_{\Gamma} p_{ij}^*(\xi, x) u_j(x) d\Gamma(x) = & \int_{\Gamma} u_{ij}^*(\xi, x) \\
p_j(x) d\Gamma(x) + \int_{\Omega} u_{ij}^*(\xi, x) b_j(x) d\Omega(x) \quad (3.22)
\end{aligned}$$

where the coefficient  $C_{ij}(\xi)$  is dependent only on the geometry of the boundary and is obtained by,

$$C_{ij}(\xi) = \delta_{ij} + \lim_{\varepsilon \rightarrow 0} \int_{\Gamma_{\varepsilon}} p_{ij}^*(\xi, x) d\Gamma(x) \quad (3.23)$$



Because of the singular nature of  $P_{ij}^*$ , the second integral on the left hand side of eqn. (3.22) is to be interpreted in the Cauchy Principal value sense. The coefficient  $C_{ij}(\xi)$  becomes equal to  $\delta_{ij}/2$  if the tangent plane at  $\xi$  is continuous; however, for practical applications, it will be seen that  $C_{ij}$  together with the corresponding principal value can be indirectly calculated by applying eqn. (3.22) to represent rigid body movement.

Neglecting body force term, the eqn. (3.22) can be written as,

$$C_{ij}(\xi) u_j(\xi) + \int_{\Gamma} P_{ij}^*(\xi, x) u_j(x) d\Gamma(x) = \int_{\Gamma} u_{ij}^*(\xi, x) P_j(x) d\Gamma(x) \quad (3.24)$$

Equation (3.24) is the starting equation for the boundary element technique. It is valid for both infinite and semi-infinite medium and also for bounded bodies as well [57].

### 3.3.2 Fundamental Solutions

Following the definition of fundamental solutions (eqn. 3.17), several different solutions can be applied in the formulation. Here, the fundamental solution due to Lord Kelvin [56] is presented which serves for a variety of problems. This corresponds to  $\Omega^*$  (Fig. 3.2) to be an infinite elastic medium and consequently,  $\Gamma^*$  is taken to

infinity.

The expressions for two-dimensional case are given by,

$$u_{ij}^* (\xi, x) = \frac{-1}{8\pi(1-\nu) G} \{ (3-4\nu) \ln(r) \delta_{ij} - r_{,i} r_{,j} \} \quad (3.25)$$

$$p_{ij}^* (\xi, x) = \frac{-1}{4\pi(1-\nu)r} \{ [ (1-2\nu) \delta_{ij} + 2r_{,i} r_{,j} ] \frac{\partial r}{\partial n} - (1-2\nu) (r_{,i} n_j - r_{,j} n_i) \} \quad (3.26)$$

where  $G$  = shear modulus,  $\nu$  = Poissons' ratio. Above expressions are valid for plane strain case and expressions corresponding to plane stress case is obtained by substituting  $\bar{\nu}$  for  $\nu$  where  $\bar{\nu} = \nu/(1+\nu)$ .

Also,  $r = r(\xi, x)$  represents distance between the load point  $\xi$  and the field point  $x$  and its derivatives are taken with reference to the coordinates of  $x$ , i.e.,

$$r = (r_i r_i)^{1/2}$$

$$r_i = x_i(x) - x_i(\xi) \quad (3.27)$$

$$r_{,i} = \frac{\partial r}{\partial x_i(x)} = r_i/r$$

Although only bounded bodies are considered so far, the validity of above fundamental solutions could easily be established in case of infinite domain [ 57] .

### 3.3.3 Numerical Implementation

Instead of attempting closed form solutions of eqn. (3.24), which is a difficult task and only attainable for simple geometries and boundary conditions, the boundary element method employs a numerical approach.

For discretization of eqn. (3.24), the boundary  $\Gamma$  is approximated by using a series of elements, the Cartesian coordinates  $x$  of points located inside of which can be expressed as,

$$x_i^e(\eta) = \sum_{l=1}^{L_e} x_i [d(e,l)] N_x^l(\eta) \quad (3.28)$$

where  $L_e$  = number of nodes of the element  $e$ ,  $d(e,l)$  = global number of node  $l$  of element  $e$ ,  $i = 1,2$  : dimension of space,  $\eta$  is the homogeneous natural coordinate associated with the element  $e$ , and  $N_x^l(\eta)$  are the shape functions [ 57], usually polynomial splines.

The BEM also assumes that the unknown variables ( $u_i$  and  $p_i$  at the boundary) are approximated in the form

$$u_i^e = \sum_{k=1}^K N_u^k u_i^k \quad (3.29)$$

$$p_i^e = \sum_{k=1}^K N_p^k p_i^k$$

where  $N_u^k$ ,  $N_p^k$  are the shape functions associated with the node  $k$  of the boundary, so that their support coincides with the surface of the element to which the point  $k$  belongs. Also,  $u_i^k$  and  $p_i^k$  are the unknowns of the problem, i.e., the unknown variables at the nodes of the boundary elements. For isoparametric elements  $N_u, N_p, N_x$  are identical and simply denoted here by  $N$ . The expressions of  $N$  for quadratic elements, which is adopted in the present application are given in Fig. 3.4.

To obtain the values of new  $M$  variables of the problem ( $M = 4 \times K$ ), it is necessary to use a numerical algorithm. For BEM, the most frequently used is the point collocation which is adopted in present application and discussed here.

Assuming, that discretization of the boundary includes  $P$  elements, the eqn. (3.24) can be expressed for a certain node  $\xi$ , by inserting eqn. (3.28) and (3.29) into eqn. (3.24), as

$$C_{ij}(\xi) u_j(\xi) + \sum_{e=1}^P \sum_{l=1}^{L_e} u_j[d(e,l)] \int_{\Gamma_e} p_{ij}^*[\xi, x(\eta)] N^1(\eta) J(\eta) d\eta \quad (3.30)$$

$$= \sum_{e=1}^P \sum_{l=1}^{L_e} p_j[d(e,l)] \int_{\Gamma_e} u_{ij}^*[\xi, x(\eta)] N^1(\eta) J(\eta) d\eta$$

where  $J(\eta)$  is the Jacobian of the application of eqn.(3.28) for each element  $e$ . By grouping some of the terms, eqn.(3.30) can be expressed as,

$$\sum_{e=1}^P \sum_{l=1}^{L_e} B_{ijel}^{\xi} u_i[d(e,l)] = \sum_{e=1}^P \sum_{l=1}^{L_e} A_{ijel}^{\xi} p_i[d(e,l)] \quad (3.31a)$$

where,

$$B_{ijel}^{\xi} = \int_{-1}^1 p_{ij}^*[\xi, x(\eta)] N^1(\eta) J(\eta) d\eta, \quad \xi \neq d(e,l) \quad (3.31b)$$

$$B_{ijel}^{\xi} = \int_{-1}^1 p_{ij}^*[\xi, x(\eta)] N^1(\eta) J(\eta) d\eta + C_{ij}(\xi), \quad \xi = d(e,l) \quad (3.31c)$$

$$A_{ijel}^{\xi} = \int_{-1}^1 u_{ij}^*[\xi, x(\eta)] N^1(\eta) J(\eta) d\eta \quad (3.31d)$$

By writing eqn. (3.31) for  $\xi = 1, 2, \dots, K$  and collecting the different terms, the system of linear equations is obtained,

$$[B] \{u\} = [A] \{v\} \quad (3.32)$$

which are  $2xK$  equations with  $4xK$  variables.

As pointed out in section 3.3.1, the coefficients  $C_{ij}$  need not be calculated explicitly and can easily be obtained by rigid body condition. Assuming unit rigid-body displacements in any one direction for a bounded body, eqn. (3.32) can be written as,

$$[B] \{I_1\} = \{0\} \quad (3.33)$$

where  $\{I_1\}$  is a vector defining a unit rigid displacement in the direction 1. Hence the diagonal terms of  $[B]$ , i.e. right hand side of eqn. (3.31c) are simply,

$$\sum_{e=1}^P \sum_{l=1}^L \delta_{\xi, d(e,l)} B_{ijel} = \sum_{e=1}^P \sum_{l=1}^L (1 - \delta_{\xi, d(e,l)}) B_{ijel} \quad (3.34a)$$

with

$$\delta_{\xi, d(e,l)} = \begin{cases} 1, & \xi = d(e,l) \\ 0, & \xi \neq d(e,l) \end{cases} \quad (3.34b)$$

For infinite or semiinfinite body, however, a further term must be added [60],

$$B_{ijel}^{\xi} = 1 - \sum_{e=1}^P \sum_{l=1}^{L_e} \delta_{\xi,d(e,l)} \quad (3.34c)$$

$$- (1 - \delta_{\xi,d(e,l)}) B_{ijel}^{\xi}$$

Once the characteristics of the program and initial data have been defined, one can then proceed to compute matrices [A],[B] in eqn. (3.32). The introduction of boundary conditions (two at each node), together with an appropriate assemblage algorithm, allow one to obtain a system of linear algebraic equations,

$$[M] \{x\} = \{F\} \quad (3.35)$$

where {x} is a vector in which all the unknowns of the problem are included. In BEM the trial functions are not locally based like FEM and so the matrix [M] is fully populated and asymmetric. This puts severe restriction on the problem size and largely dilutes the advantage gained by avoiding the discretization of domain and thereby reducing number of variables. This has resulted in many techniques to solve this type of problem [59]. However, if one ultimately treats the BEM region as one of the finite elements by a proper coupling

strategy, one can use the standard finite element equation solution technique such as a frontal solver.

In order to get the system of eqn. (3.35), it is necessary to compute a set of coefficients whose expressions are given in eqn. (3.31). This entails integrating some functions which depend on the coordinates of the collocation and integration points. This is extremely difficult business in BEM and is not as straight forward as in FEM. Generally Gaussian integration scheme is used to compute the coefficients, though problem arises due to singularity of the integrals because of the basic singular nature of the fundamental solutions. Although this singularity is saved theoretically by considering the integrals in its Cauchy principal value sense, it is obvious that a standard Gaussian quadrature process could provide inaccurate results when the distance between collocation point  $\xi$  and the integration point  $x(n)$  on the element  $e$ , over which the integration is carried out, tends to zero, and, more over, when  $\xi$  belong to  $e$ , so that the functions to be integrated are not bounded.

Though various sophisticated techniques could be used to circumvent this problem such as analytic and semi-analytic integration, adaptive numerical integration etc. [60] one simple technique [60] is to use special numerical integration scheme for singular element and standard Gaussian quadrature for the rest of the elements with increasing number of integration points as the distance between collocation point and integration point reduces.



For elements close to the collocation point, a rational approach is taken as given below:

$$\begin{aligned}
 S &\leq 1.5 \rightarrow Q = 12 \\
 1.5 &\leq S \leq 5.5 \rightarrow Q = 8 \\
 5.5 &< S \rightarrow Q = 6
 \end{aligned}
 \tag{3.36}$$

in which  $S$  is the distance between the centre of the element and singular node  $\xi$  divided by the distance between the 1st and 3rd node of the element and  $Q$  is the number of Gauss points. This criterion was found to be accurate enough without losing computational efficiency, even though it is not acceptable for severely distorted elements.

For the special case of the singular node being coincident with one of the element nodes, the coefficients of matrix  $[B]$  remains non-singular except for the  $2 \times 2$  diagonal submatrix which corresponds to the singular node. This is because, even though there is a singularity of type  $(1/r)$  in  $p_{ij}^*$ , the shape functions  $N(\eta)$  are of order  $r$  and hence their product tend to a finite limit as  $r \rightarrow 0$ . For determining the singular leading diagonal submatrix, rigid body movement (eqn. 3.34), is used obviating the need to go for complicated schemes to compute coefficient  $C_{ij}(\xi)$  and the principal value integral.

The calculation of coefficients of matrix  $[A]$  is more difficult for singular elements as singularity of type  $\ln(1/r)$  is involved. The terms which do not have this kind of singularity are calculated with standard Gaussian quadrature scheme with increasing Gauss points according to scheme given by eqn. (3.36). The terms involving  $\ln(1/r)$  singularity are transformed by writing,

$$\ln(1/r) = \ln(\alpha/r) + \ln(1/\alpha), \quad (3.37a)$$

so that the integral containing  $\ln(1/r)$  may be written as,

$$\int_{-1}^1 \ln\left(\frac{1}{r(\eta)}\right) d\eta = \int_{-1}^1 \ln\left[\frac{\alpha}{r(\eta)}\right] d\eta + b \int_0^1 \ln\left(\frac{1}{\alpha}\right) d\alpha \quad (3.37b)$$

where  $b$  is the Jacobian arising out of the linear transformation  $\eta$  to  $\alpha$ . In eqn. (3.37b),  $\alpha$  is chosen such that  $\alpha \rightarrow 0$  as  $r(\eta) \rightarrow 0$  thus making the 1st integral on the right hand side regular. The second integral on right hand side has got integrable singularity and can be integrated using logarithmic Gaussian quadrature formula [58].

### 3.3.4 Stresses and Displacements Inside the Body

Once the values of displacements and tractions are known on the boundary, one can calculate the displacements at

any internal point directly from Somigliana's identity, eqn.(3.21). For calculating stresses, the eqn. (3.21) is differentiated at the internal points, to obtain,

$$\sigma_{ij}(\xi) = \int_{\Gamma} u_{ijk}^*(\xi, x) p_k(x) d\Gamma(x) - \int_{\Gamma} p_{ijk}^*(\xi, x) u_k(x) d\Gamma(x) \quad (3.38)$$

where, for the Kelvin fundamental solution, the new tensors are [57],

$$u_{ijk}^*(\xi, x) = \frac{1}{4\pi(1-\nu)r} \{ (1-2\nu) (r_{,j} \delta_{ki} + r_{,i} \delta_{jk} - r_{,k} \delta_{ij}) + 2r_{,i} r_{,j} r_{,k} \} \quad (3.39a)$$

$$\begin{aligned} p_{ijk}^*(\xi, x) = & \frac{G}{2\pi(1-\nu)r^2} \{ 2 \frac{\partial r}{\partial n} [ (1-2\nu) \delta_{ij} r_{,k} + \nu (\delta_{ik} r_{,j} + \delta_{jk} r_{,i}) - 4r_{,i} r_{,j} r_{,k} ] \\ & + 2\nu (n_i r_{,j} r_{,k} + n_j r_{,i} r_{,k}) + (1-2\nu) (2n_k r_{,i} r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) - (1-4\nu) n_k \delta_{ij} \} \end{aligned} \quad (3.39b)$$

### 3.3.5 BEM for Cracked Bodies

If a boundary element method is applied directly to a non-symmetrical crack, the resulting boundary equations have a singular matrix [61] , since the method can not distinguish between two surfaces in the same plane. Several strategies have been proposed to overcome this difficulty such as representing the crack as notch, symmetric crack modelling, use of special Green's function and flat crack modelling. All of them suffer from some inherent drawbacks which prohibit their use in general mixed mode cases. An extensive discussion regarding is given by Cruse [61].

An alternative strategy has been proposed by Blanford et . al. [ 18] in which multidomain discretization using isoparametric quadratic boundary elements are used (Fig. 3.5). As the two crack surfaces are modelled in two separate domains, so the problem of two displacement variables along the crack and non-uniqueness of boundary integral equation resulting from crack surface loading do not arise. The prescribed tractions and unknown displacements along the crack surface are easily represented independently in each domain. Further more, the crack representation is mathematically correct, i.e. the crack is not represented as a notch.

To properly represent the analytically dominant  $\sqrt{r}$  and  $1/\sqrt{r}$  behaviour exhibited by the displacements and tractions

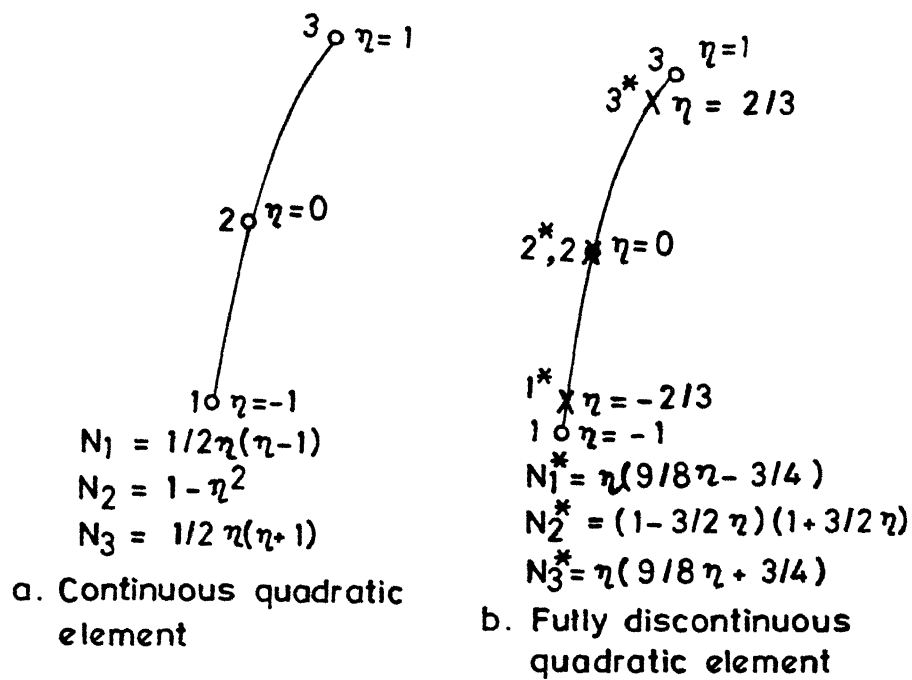


FIG.3.4 Different types of quadratic boundary elements

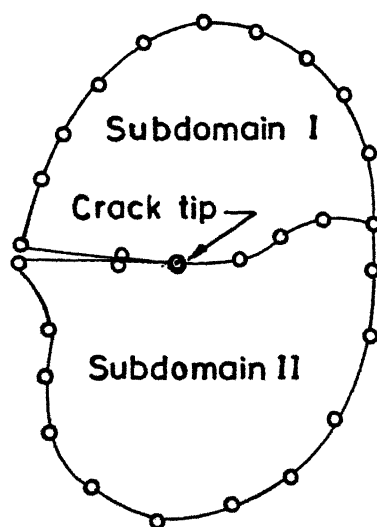


FIG. 3.5 Multidomain discretization of cracked body

CENTRAL LIBRARY  
105914

respectively in the vicinity of crack tip, the classical quadratic boundary elements has to be transformed. It has been shown in finite element literature [62], that the  $\sqrt{r}$  displacement variation can be represented using the isoparametric quadratic element by placing the mid point node at the quarter point, as shown in Fig. 3.6. Using the shape functions defined in Fig. 3.6, the variation of the displacements and tractions then become,

$$\begin{aligned} u_i \\ p_i \end{aligned} = A_i^1 + A_i^2 \sqrt{r} + A_i^3 r, \quad i = 1, 2 \quad (3.40)$$

In finite element applications, the tractions are obtained by differentiating the displacement interpolation functions, which results in correct  $1/\sqrt{r}$  singularity in the traction variation. However, in BEM, the displacement and traction variations are independently represented. The inclusion of  $1/\sqrt{r}$  singularity in traction interpolation is accompanied by multiplying the right hand side of eqn. (3.40) by  $\sqrt{1/r}$  [18]

$$p_i = (A_i^1 + A_i^2 \sqrt{r} + A_i^3 r) \sqrt{1/r} = B_i^1 / \sqrt{r} + B_i^2 + B_i^3 \sqrt{r} \quad (3.41)$$

which corresponds to the correct form;  $l$  in the above equation is the length of the crack tip element and the expression  $\sqrt{l/r}$  can be written in non-dimensional coordinate form as shown in Fig. 3.6.

The traction singular quarter point element described above is extremely useful to model crack tip boundaries. They can be used for mixed mode problem of general nature and can be utilized to extract SIF's easily.

Although several strategies could be adopted to arrive at the stress intensity factors, such as J-integral, strain energy release rate etc, one particular method which is computationally efficient, gives accurate results, requires no interpolation and can be applied in general mixed mode problems is the displacement correlation method [18], [63]. Also quarter point element fits into this scheme directly.

Denoting the displacements along the crack axis as  $u$  (crack sliding displacement, CSD) and the displacements normal to the crack axis as  $v$  (crack opening displacements, COD), the variation of CSD and COD are, using eqn. (2.2)

$$\begin{aligned} u' &= u'_A + (-3u'_A + 4u'_B - u'_C) \sqrt{r/l} + (2u'_A + 4u'_B + 2u'_C) r/l \\ v' &= v'_A + (-3v'_A + 4v'_B - v'_C) \sqrt{r/l} + (2v'_A - 4v'_B + 2v'_C) r/l \end{aligned} \quad (3.42)$$

where the prime indicates that the global coordinate nodal displacements have been transformed to the crack tip coordinate system defined in Fig. 3.7.

For a symmetric crack problem (Mode I).  $u' = u'_A = 0$  and eqn. (3.42) reduces to,

$$v' = (4v'_B - v'_C) \sqrt{r/l} + (-4v'_B + 2v'_C) r/l \quad (3.43)$$

Equating the coefficients of  $r$  in eqn. (3.43) to that of  $u_2$  in eqn. (2.2) and taking  $\theta = 180^\circ$ ,

$$K_I = \frac{2G}{k+1} \sqrt{\frac{2\pi}{l}} (4v'_B - v'_C) \quad (3.44)$$

where  $k = 3 - 4\nu$  for plane stress and  $k = (3-\nu)/(1+\nu)$  for plane strain.

The general mixed mode case is shown in Fig. (3.7b) and for  $\theta = 180^\circ$ , interaction of  $K_I$  and  $K_{II}$  in eqn. (2.2) is decoupled, resulting in,

$$K_I = \frac{G}{k+1} \sqrt{\frac{2\pi}{l}} [4(v'_B - v'_D) + (v'_E - v'_C)] \quad (3.45)$$

$$K_{II} = \frac{G}{k+1} \sqrt{\frac{2\pi}{l}} [4(u'_B - u'_D) + (u'_E - u'_C)]$$

Algorithmically, to get the SIFs, the displacements and coordinates of crack face nodes belonging to the quarter point elements need to be flagged, and, retrieved, for each crack increment solution. These are then transformed to a simple subroutine which codes eqn. (3.44) and eqn. (3.45).



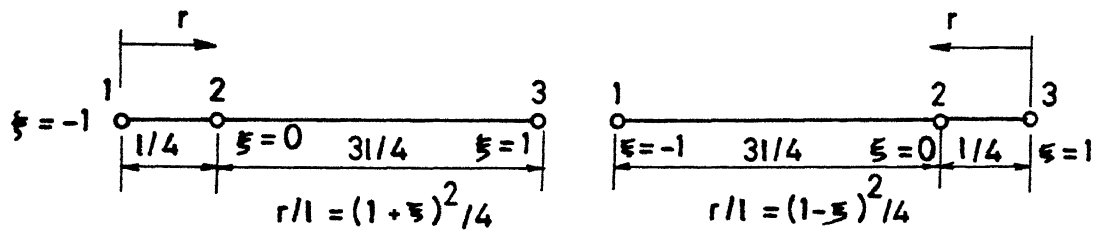


FIG. 3.6 Traction singular quarter point boundary element

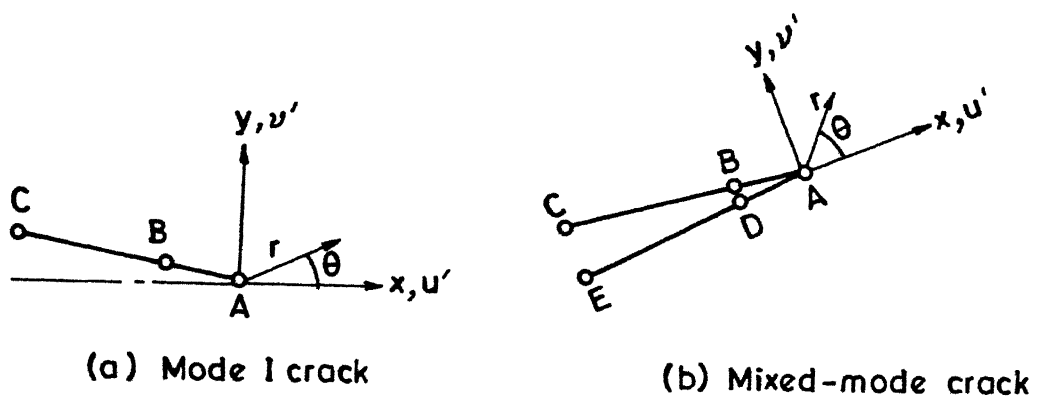


FIG. 3.7 Element geometries for SIF computation

### 3.4 Coupling of BEM and FEM

The most commonly used technique to couple the two methods is to transform the boundary element region into an equivalent finite element. This has two inherent problems, namely,

- i) The stiffness matrix thus formed is nonsymmetric.
- ii) The equilibrium is not identically satisfied.

In order to overcome these difficulties, several methods have been proposed [24] , [25 ]. The energy approach given by Kelly et.al. [24 ] is one of the most attractive and conceptually simple.

#### 3.4.1 Energy Approach

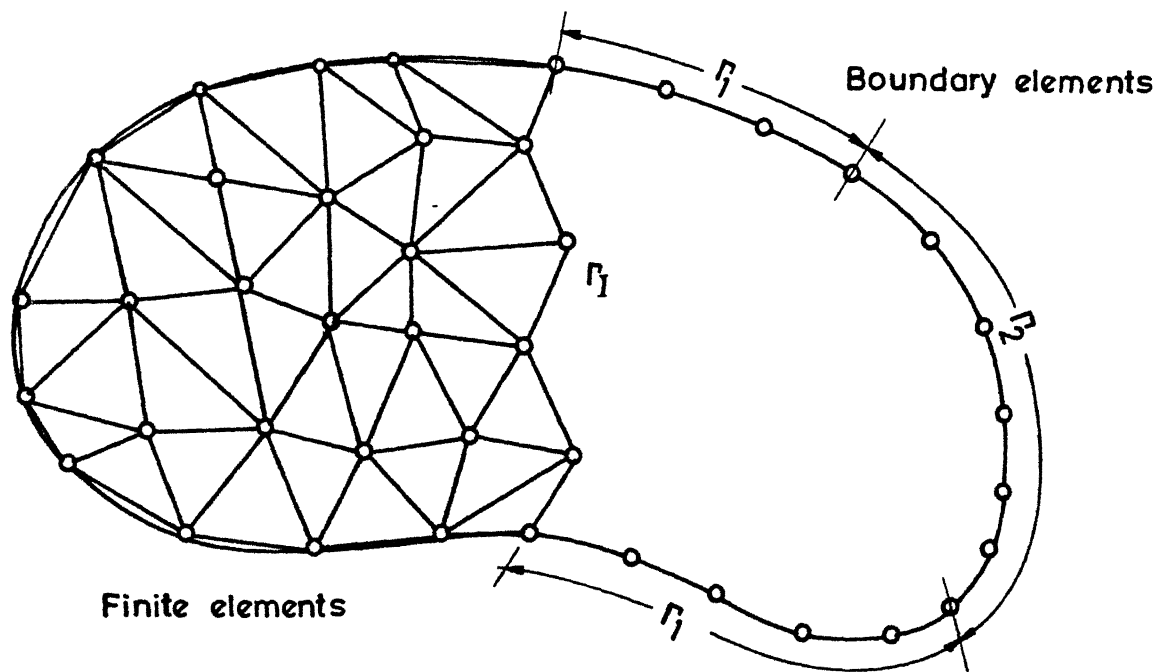
To generate a symmetric system of equations from the direct boundary integral procedure, one has to minimise the energy functional of the form, for the boundary element region in Fig. 3.8,

$$\Pi = \frac{1}{2} \int_{\Gamma} \{p\}^T \{u\} \, d\Gamma - \int_{\Gamma_2} \{u\}^T \{\bar{p}\} \, d\Gamma \quad (3.46)$$

If the boundary interpolant for  $\{u\}$  and  $\{p\}$ , which are of the form.

$$\{u\} = [N] \{\bar{u}\} \quad (3.47)$$

$$\{p\} = [M] \{\bar{p}\}$$



**FIG. 3.8** Domain divided into finite element and boundary element regions

are substituted into eqn. (3.46),  $\pi$  becomes,

$$\pi = \frac{1}{2} \{ \bar{p} \}^T \int_{\Gamma} [M]^T [N] d\Gamma \{ \bar{u} \} - \{ \bar{u} \}^T \int_{\Gamma} [N]^T \{ \bar{p} \} d\Gamma \quad (3.48)$$

The boundary collocation around the boundary element region gives direct boundary integral system of equations (3.32) in which rigid body condition is implicitly satisfied by means of eqn. (3.33). In order to ensure equilibrium condition, eqn. (3.32) is supplemented with an equilibrium equation

$$\int_{\Gamma} \{ p \} d\Gamma \equiv \int_{\Gamma} [M(\Gamma)] d\Gamma \{ \hat{p} \} = \{ 0 \} \quad (3.49)$$

Linking the direct boundary integral eqn. (3.32) with the equilibrium eqn. (3.49), one obtains,

$$\begin{bmatrix} [A] \\ [0] \end{bmatrix} \begin{Bmatrix} \{ u \} \\ \{ \lambda \} \end{Bmatrix} = \begin{bmatrix} [B] & [0] \\ [0]^T & [0] \end{bmatrix} \begin{Bmatrix} \{ p \} \\ \{ \lambda \} \end{Bmatrix} \quad (3.50)$$

where  $\{ \lambda \}$  takes the role of a Lagrange multiplier and matrix  $[0]$  is given by,

$$[0] = \int_{\Gamma} [M(\Gamma)] d\Gamma \quad (3.51)$$

The role of the Lagrange multiplier in eqn. (3.50) is therefore to introduce a controlled error into each of equations (3.32)

which can no longer be satisfied identically if the extra condition (3.49) is imposed.

Inverting eqn. (3.50) and partitioning, one can express nodal tractions in terms of nodal displacements,

$$\{p\} = [E] \{u\} \quad (3.52)$$

Putting eqn. (3.52) into eqn. (3.48), the functional  $\pi$  can be expressed as,

$$\pi = \frac{1}{2} \{\bar{u}\}^T [E]^T \int_{\Gamma} [M]^T [N] d\Gamma \{\bar{u}\} - \int_{\Gamma} [N]^T \{\bar{p}\} d\Gamma \{\bar{u}\}^T \quad (3.53)$$

The minimization of  $\pi$ , which requires  $\partial \pi / \partial \bar{u} = 0$  leads to

$$[k'] \{u\} + \{F'\} = \{0\} \quad (3.54)$$

where

$$[k'] = \frac{1}{2} [E]^T \int_{\Gamma} [M]^T [N] d\Gamma + \frac{1}{2} [[E]^T \int_{\Gamma} [M]^T [N] d\Gamma]^T \quad (3.55)$$

and

$$\{F'\} = - \int_{\Gamma_2} [N]^T \{\bar{p}\} d\Gamma \quad (3.56)$$

The matrix eqn. (3.54) is of the same form as the finite element stiffness equations (3.12), and so, the matrices  $[K']$  and  $\{F'\}$  can be assembled into standard finite element systems as contributions from a new element. Compatibility

between elements is ensured if the boundary interpolants for the boundary integral region are identical to the interpolation functions of the adjacent finite elements.

As pointed out in [64] , special care should be taken for corner nodes and at nodes having traction discontinuity, which cannot be taken into account by a classical discretization by ordinary continuous elements. One method of solving this is to displace the nodes from the element corners, i.e. using discontinuous elements [65] where nodal values of tractions are chosen inside the element as shown in Fig. 3.4. However, such a discretization increases the number of traction nodal values and consequently leads to a non-square system of equations and, therefore, a non-solvable one.

To overcome this problem, the collocation procedure is modified such that collocation points coincide with the traction nodal value positions. Also, in order to use the integration scheme described in section 3.3.3, displacement freedom nodes are also made to coincide with this discontinuous nodes. Then after computing matrices  $[B^*]$  and  $[A]([B^*])$  correspond to displacements  $\{u^*\}$  - Fig. 3.4) and in order to keep compatibility with a possible finite element zone, one must return to the previous interpolation scheme i.e. that of continuous nodal displacement scheme. This is simply achieved by expressing the inside nodal values  $\{u^*\}$  in terms of end nodal values  $\{u\}$  ,

$$\{u^*\} = [s]\{u\} \quad (3.57)$$

The main disadvantage of this coupling strategy is that one inversion is required (eqn. 3.52). But the other possibilities reported in literature [27] require solution of set of boundary equations with multiple right hand side, which does not produce any significant amount of saving. An error analysis of different coupling strategies has been reported in [25], where it is shown that the coupling strategy outlined in this section gives better results than the other competing techniques.

#### 3.4.2 Advantages of Coupling BEM and FEM

A combined application of both BEM and FEM becomes attractive as each method can be applied in the area where it is best suited. By such coupling the following advantages of BEM are exploited in a linear elastic fracture mechanics problem by using boundary elements around the crack tip:

1. The inherent advantage of BEM is that all approximations are confined to the boundary. Thus it gives better results in regions of high stress gradient such as a crack tip.
2. The ease of the use of singular elements to reproduce the crack tip singularity. Traction singular quarter point elements are very easily and efficiently generated from ordinary boundary elements and as no domain subdivision is required around the crack tip, so the solution becomes less mesh dependent.

3. The ease of discretizing a moving crack boundary. The error associated with crack propagation analysis of FEM due to irregular discretization around a crack tip is avoided by using boundary elements around a crack tip giving a better estimation of the crack path.
4. The need to discretize the boundary only in boundary element region. This reduces number of nodes and elements in the problem and so considerable saving in mesh generation, mesh editing, remeshing timings is achieved.

On the other hand, use of finite elements away from the crack tip offers following advantages:

1. Better efficiency of FEM in analysis of finite body problems; for such analysis even quadratic boundary elements are sometimes less efficient than the FEM [66].
2. Use of localized discretization, i.e. locally based functions which produce narrowly banded final set of equations. This contrasts with BEM in which fully populated unsymmetric matrices are generated which largely dilutes the advantage gained in BEM by discretizing boundary only.



3. Availability of efficient large number of algebraic equations solution algorithm. Frontal solver is utilised in present application which requires very little core storage and also it has resolution capability in that it can give solutions for many load vectors without recalculating and reassembling the stiffness matrices again and again. This is particularly useful to determine very quickly and efficiently the effect of varying load on a propagating crack.

## CHAPTER 4

### INTERACTIVE SIMULATION OF CRACK PROPAGATION

#### 4.1 Introduction

As apparent from discussions of previous chapters, crack propagation is a highly complicated and geometrically non-linear phenomenon. Because of the complexity of the modelling of cracks, it is currently impossible to create a general "problem in, solution out" or "expert system" computer code to model automatic crack propagation. Instead, by increasing the interactivity of the code, its flexibility is increased so that user learns much more about cracking behavior than he or she could with a 'problem in, solution out' type of code.

A necessary component of an interactive simulation environment is computer graphics and related concepts. A computer graphics subsystem appropriate for simulation of complex fracture events must involve more than simple two-dimensional plots, charts and mesh displays. There must be capabilities for generating and editing mesh, interactively changing problem attributes such as loads, displacement boundary conditions and an automatic remeshing capability which modifies the mesh to reflect the crack extension.

To achieve all these, the program must use a robust data structure supported by intelligent algorithms which perform fast queries and manipulate data base.

This chapter outlines the mesh generation technique in section 4.2 and data base design in section 4.3. Lastly, the program integration aspect to achieve all the capabilities needed for a fracture propagation simulation, namely, the FEM and BEM, fracture mechanics and interactive graphics concepts is discussed in section 4.4.

## 4.2 Mesh Generation

The blending function approach [38] , [ 68 ] is currently most popular because of its ability to produce well conditioned meshes for very complex boundaries. One such blending function method, called transfinite mapping method is probably the most powerful of all mapping methods. It has been adopted in the present application because it can be used for generation of both finite and boundary element mesh in a unified manner.

### 4.2.1 Transfinite Mapping Method

Let the problem region  $R$  be an open, bounded and simply connected planer region in  $E^2$  and  $S$ , an open unit square  $(0,1) \times (0,1)$  in  $R^2$ . Let there be a one-to-one (univalent) mapping  $\vec{U} : S \leftrightarrow R$  such that each point  $(x,y) \in R$

has a unique correspondent  $(s,t) \in S$ . In other words, goal here is to achieve a curvilinear coordinatization of  $R$  such that each point  $(x,y)$  in  $R$  may be uniquely referenced by its generalized coordinate  $(s,t)$ .

A canonical domain (canonical geometry) means a standard domain, such as a rectangle, triangle, circle etc. for which a natural (curvilinear) coordinatization is known. If the given problem domain  $R$  is reminiscent of some canonical domain, the objective will be to coordinatize  $R$  with a generalized system analogous to that of the canonical-region. To do so, one widely applicable method can be used (developed for mathematical description of free-form shapes in two and three-dimensions), called "blending functions" methods [69]. In these, a  $n$ -dimensional manifold is expressed as weighted ("blended") combination of manifolds of dimensionality  $n-1$  or less.

Let  $\vec{F}$  be a continuous vector valued primitive function which maps  $S$  invertibly onto  $R$ .  $\vec{F}$  is an unknown function and one can identify curves in  $R$  as being the images under  $\vec{F}$  of constant coordinate lines in  $S$ . Also, let  $P$  be a simple projector such that the vector-valued approximation,  $P[\vec{F}]$  is a continuous and univalent map of  $S$  onto  $R$ .

On  $S$ , consider a cartesian product partition  $\pi_s \times \pi_t$  where  $\pi_s : 0 = S_0 < S_1 < \dots < S_M = 1$  and  $\pi_t : 0 = t_0 < t_1 < \dots < t_N = 1$ ,

and let  $\{\phi_i(s)\}_{i=0}^M$  and  $\{\psi_j(t)\}_{j=0}^N$  be functions satisfying the cardinality property.

$$\phi_i(s_k) = \delta_{ik}, \quad \psi_j(t_1) = \delta_{j1} \quad (4.1)$$

where  $\delta_{ij}$  is the Kronecker delta.

Further, define the projectors  $P_s$  and  $P_t$  by,

$$P_s[\vec{F}] = \sum_{i=0}^M \vec{F}(S_i, t) \phi_i(s) \quad (4.2)$$

$$P_t[\vec{F}] = \sum_{j=0}^N \vec{F}(S, t_j) \psi_j(t)$$

then the product projector (tensor product interpolant),

$$P_s P_t[\vec{F}] = \sum_{i=0}^M \sum_{j=0}^N \vec{F}(S_i, t_j) \phi_i(s) \psi_j(t) \quad (4.3)$$

interpolates  $\vec{F}$  at the  $(M+1)(N+1)$  points  $\{(S_i, t_j)\}_{i=0, j=0}^{M, N}$ .

In contrast, the transfinite interpolant, defined by

Boolean Sum,

$$P_s \oplus P_t[\vec{F}] \equiv P_s[\vec{F}] + P_t[\vec{F}] - P_s P_t[\vec{F}] \quad (4.4)$$

interpolates to  $\vec{F}$  at all points along the two sets of lines  $\{S = S_i\}_{i=0}^M$  and  $\{t = t_j\}_{j=0}^N$ . The functions  $\phi_i$  and  $\psi_j$  are

called blending functions.

If the blending functions are chosen as the Lagrange polynomials (of minimal degree) satisfying eqn. (4.1), then  $P_s \oplus P_t [\vec{F}]$  is the transfinite Lagrange interpolant. In particular, if  $M = N = 1$ , then  $P_s \oplus P_t [\vec{F}]$  is the bilinearly blended transfinite Lagrange interpolant,

$$\begin{aligned} P_s \oplus P_t [\vec{F}] = & (1-s) \vec{F}(0,t) + s \vec{F}(1,t) + (1-t) \vec{F}(s,0) \\ & + t \vec{F}(s,1) - (1-s)(1-t) \vec{F}(0,0) \\ & - (1-s)t \vec{F}(0,1) - s(1-t) \vec{F}(1,0) - st \vec{F}(1,1) \end{aligned} \quad (4.5)$$

The images of lines  $s = \text{constant}$  and  $t = \text{constant}$  in  $S$  are curves in  $R$ . In particular if boundary of  $R$ ,  $\partial R$  is identified as being comprised of the four curve segments  $\vec{F}(0,t)$ ,  $\vec{F}(1,t)$ ,  $\vec{F}(s,0)$ , and  $\vec{F}(s,1)$ , then  $\partial S \rightarrow \partial R$  for any choice of functions  $\{\phi_i\}$  and  $\{\psi_j\}$  which satisfy eqn. (4.1). Higher order interpolants may be used to force the co-ordinate curves to pass through specified curves on the interior of  $R$  [68].

The general transfinite mapping describes an approximate surface or volume which matches a desired or true surface or volume at a non-denumerable set of points. It is this property

which gives rise to the term transfinite mapping. This property contracts with the isoparametric mappings [ 74 ] which match the true surface at only a finite number of points. So transfinite mapping can be made to exactly model all region boundaries and no geometric error is introduced by the mapping.

Two classes of representation of the boundary of a region is possible, namely, continuous form and discrete form. The continuous forms represent the position vector to a boundary curve as a function of some parametric co-ordinate. The discrete representation [ 38] consists of finite lists of points located on the curve, with a unique co-ordinate associated with each point in the list. The discrete form of representation is completely general and the present precessor utilises this concept,

#### 4.2.2 Implementation

Discrete transfinite mapping [38] may be used to map any mesh in the appropriate primitive polygon to the actual region. Thus any mesh topology may be used. However, it is convenient to sacrifice topological generality by choosing the intersections of the constant coordinate curves for use as nodal points in order to minimize input. In bilinear mapping, s direction curves are defined by constant t coordinate values and vice-versa. Thus a list of n constant

$t$  values  $t_j$ ,  $j = 1, n$  defines  $n$  curves running in  $s$  direction and a list of  $m$  constant  $s$  values  $s_i$ ,  $i = 1, m$  defines a set of  $m$  curves running in  $t$  direction. Examination of eqn.(4.5) reveals that to evaluate the location of  $nm$  intersection points of these curves, the boundary curves need to be evaluated at  $2(n+m)$  points:

$$\{\vec{F}(0, t_i), \vec{F}(1, t_i)\} \quad i=1, n \quad \text{and} \quad \{\vec{F}(s_j, 0), \vec{F}(s_j, 1)\} \quad j=1, m$$

For efficient evaluation of all  $nm$  intersection points,  $2(n+m)$  boundary curve points are evaluated first and stored to avoid redundant calculations.

An apparent drawback of the discrete form of curve representation is the large amount of data required to describe a boundary curve in discrete form. This drawback is eliminated by providing a package of curve generator that automatically create discrete curve descriptions based on various continuous models. Each curve is assigned a unique curve number and list of discrete points representing it is stored in separate arrays. This allows user input to requirement to be reduced to a minimum.

When highly distorted boundary curves are used to define a region, a problem of overspill may result [38]. This problem occurs when finite element mesh lines cross



outside the region boundaries. Higher order mapping may be used to force the mesh lines to pass through some predefined constraint curves on the interior of the region [ 69] . In the present preprocessor, this problem is overcome by breaking the highly distorted region down into two or more regions with more regular geometry. Appropriate book keepings are maintained to avoid generation of duplicate nodes along the curves shared by more than one region. Regions are first reordered so that regions sharing a common point may be numbered in a continuous sequence. If there is still any duplicate node, the node numbers are compressed and the node numbers associated with individual elements are corrected.

In order to apply a uniform algorithm for generating both boundary and finite element mesh, regions which are to be discretized by boundary elements are treated as a 'super' finite element, i.e. a big finite element with more than eight nodes. These regions are flagged and while transfinite mapping is applied to generate nodes, the internal (domain) node generation are suppressed for such regions, so that only boundary is discretized. A finite element type node-element connectivity for the super finite element is then generated and this is later translated into a boundary element type node-element connectivity by appropriate coding.

### 4.3 Data Base Design

The program data base is the Kernel which ties the rest of the program together and it has a profound impact on the design of the rest of the program. The data base used in the present implementation is designed around a winged edge data structure, introduced by Baumgart [41] to describe the surface topology of polyhedra.

#### 4.3.1 Definitions

At this point a few definitions are necessary.

Topological information is the adjacency description of a polyhedron. In finite element sense, this would be information such as which nodes are adjacent to a given element or which elements are adjacent to a given node.

Geometric information, in contrast, is information about actual point locations (i.e. the nodal coordinates in a finite element mesh).

A vertex is a unique point in space; in planer case, a point in the plane in which the mesh construction will take place. In the discussion to follow, a vertex is isomorphic to a node in finite element sense and the terms will be used interchangeably.

An edge or arc is a set of two vertices. The edges discussed here will be directed edges, i.e. edge points from one vertex to the other. The finite element analogy to the edge is a portion of an element boundary that connects two adjacent elements.

A graph is a set of vertices and distinct edges which utilize the vertices. One can view finite element mesh as a graph. A graph that allow self loops (i.e. graph configuration in which an edge joins a vertex to itself) and multiple edges (or multigraph i.e. a graph configuration where multiple edges are allowed to join the same two vertices) are called pseudographs. When graph is referred to here, it is not referring to pseudo graphs.

A graph can be embedded (or mapped) onto a surface if it is drawn on the surface so that no two edges intersect, except at their incident vertices.

Faces are the regions of a surface defined by a graph embedded on a surface. Each face is a connected component of the set obtained by subtracting the vertices and edges of the embedded graph from the surface. The boundary of a face consists of those edges and vertices of the embedded graph whose every part touches upon the face. A simply connected face has a single, connected boundary. A multiply connected face has boundary that consists of two or more disconnected components, as in a face with a

hole in it. Here, only simply connected faces are referred to.

Finite element mesh is a collection of faces, i.e. all the elements in a finite element mesh will be faces. There is also one special face that arises from the process of mapping a polyhedron on to a sphere (to be described in the next section). The face that covers all area exterior to the mesh must be stored explicitly with the rest of the faces. For such a collection to define a valid mesh, the following must be satisfied [70] :

- (1) Faces of the collection may intersect only at common edges or vertices (if at all)
- (2) Each of the edges is shared by exactly two faces
- (3) Faces around each vertex form a single circuit, i.e. they can be arranged in cycle, such that each pair of consecutive faces in the cycle meet at an edge adjacent to the vertex.

#### 4.3.2 Winged Edge Data Structure Theory

At a first glance, it is not obvious that a data structure developed to store the topology of a polyhedra would be useful for representing a two-dimensional finite element mesh. However, the topology of a polyhedra can always be mapped onto the surface of a sphere (Fig. 4.1).

The mapping can be performed in such a way that one of the polygons covers a very large percentage of sphere's surface. Attention can be restricted to the portion of the surface that contains all other polygons. This essentially is the topology of the polyhedron mapped on to a plane, provided the curvature of the sphere's surface is sufficiently small. Finite element analysis is concerned with this planer topology; the polyhedron that is represented by it is not important [47].

The winged edge polyhedron representation contains three sets of records: one set each for vertices, edges and faces. Each record contains information associated with the topological entity it represents and also contains pointers to other records in the data base (Fig. 4.3). It is by means of these pointers to other records that the adjacency information inherent in the data structure can be extracted. The winged edge data structure and similar representation are called edge based data structures [44], because the edge records are used as keys to access other type of records. The edge records are used for this function because the quantity of adjacency information associated with an edge is invariant from edge to edge. That is, each edge is always adjacent to exactly two faces and exactly two vertices, whereas a face or a vertex can be adjacent to any number of edges (Fig. 4.2).

As seen in Fig. 4.2, the winged edge structure maintains the adjacency information with pointers to two faces,

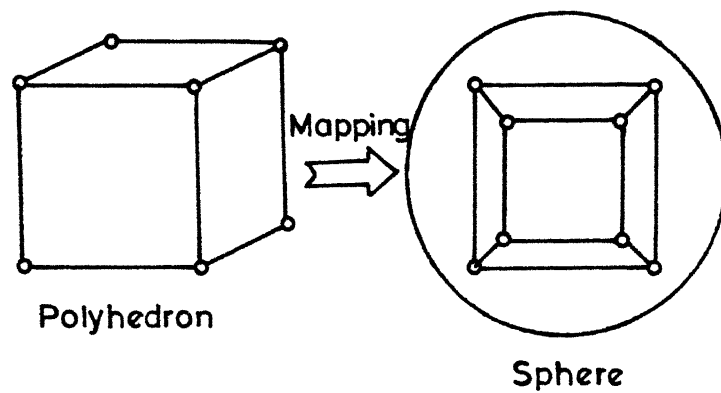


FIG.4.1 Mapping of polyhedron (in this case a cube ) on to the surface of a sphere

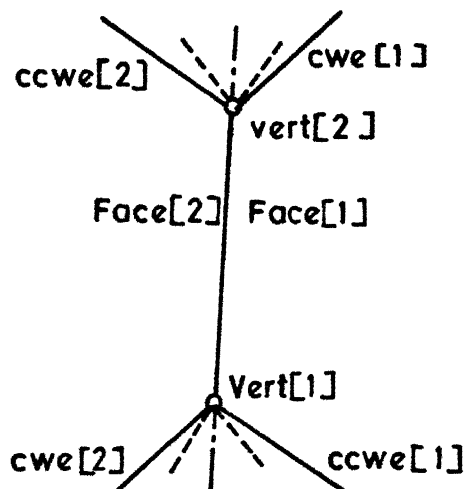


FIG.4.2 The winged-edge data structure

the two vertices and some of the edges adjacent to the reference edge. This last set of adjacencies is divided into two sections, each associated with the use of one of the sides of the reference edges in the circuit of edges around a face. The only edge adjacencies represented, therefore, are the four edges that directly follow or precedes the reference edge in edge cycles surrounding the two faces adjacent to the reference edge. Thus, the cwe and ccwe field names used here refer to their use in determining the cycle of edges surrounding a face, as viewed from outside the plane looking towards the surface. Finally, Neptr and Peptr are the forward and backward pointers for a doubly linked list that contains all the edge records (Fig.4.3a).

Figure (4.3b) shows the structure for a vertex record. The vertex records not only contain adjacency information but they also contain geometric information (i.e. nodal coordinates) and information used in the finite element processing. The Eptr field points to one of the edges to which the vertex is adjacent. The Neptr and Peptr fields are forward and backward pointers to a doubly linked vertex list similar to the edge list. The vertex record also contains global finite element node number associated with the vertex. This information is used while assembling the finite element equations to interpret the structural response information.

The structure for a face record is shown in Fig. (4.3c). The record contains the pointers to the face list and a pointer

Neptr	Peptr
Face [ 1 ]	Face [ 2 ]
Vert [ 1 ]	Vert [ 2 ]
Cwe [ 1 ]	CWe [ 2 ]
CCWe [ 1 ]	CCWe [ 2 ]

a. Edge record

Nvptr	Pvptr
Epтр	
X-coordinate	
Y-coordinate	
Node Number	

b. Vertex record

N face	P face
Face Number	Epтр
Max. X-coordinate	
Max. Y-coordinate	
Min. X-coordinate	
Min. Y-coordinate	
Face Type	Material

c. Face record

Fig. 4.3 : Data Structure Record Organisation



to one of its adjacent edges. In addition to the adjacency information, the record also contains bounding area information for the face. This information is used to speed up the process of determining if a given point is or is not within a face. If the point is not within the bounding area of the face, the face can be rejected and no further processing is necessary. The face record also contains a face-type field. This field tells whether the face is being used to represent a finite element, and if so, what type. The two remaining fields are only defined if the face is representing a finite element. The face-number field contains an index into data structure that maintains all the stiffness matrices of all the elements in the mesh. The material field identifies the elements' material type.

#### 4.3.3 Euler Operators

The main problem of boundary representation type solid modeller, for which W-E data structure was first introduced, is their complexity; models of practically interesting objects consists of tens or hundreds of faces, edges and vertices. Consequently, their construction is an elaborate and error-prone task.

Euler operators [42] were first introduced in order to simplify the description of boundary representation. Their

key idea is that the construction of each meaningful boundary representation can take place in a step wise fashion using a small set of data structure constructors which effectively hide to details of representation used.

Consider a simple polyhedron with  $V$  vertices,  $E$  edges and  $F$  faces. Then, according to Euler formula,

$$V - E + F = 2 \quad (4.6)$$

Euler operators derive their name from this formula. Following Braid et.al. [42], eqn. (4.6) can be interpreted as the equation of a two-dimensional hyperplane in the three dimensional lattice axes  $\langle V, E, F \rangle$ . Obviously, the two-dimensional plane can be spanned by a collection of two-linearly independent vectors of the hyperplane satisfying eqn. (4.6). Euler operators then may be viewed as such vectors, which add or remove vertices, edges, faces as determined by the components of the corresponding vector of the plane given by eqn. (4.6). List of operators chosen for the present implementation are given in Table 4.1.

The semantics of Euler operators are best represented by plane models. Only informal descriptions of each operators are given with reference to Fig. 4.4; rigorous formal

Table 4.1 : Euler Operators

Operator	Transition vector (V,E,F)	Explanation
MSFV (F, V : OUT)	1, 0, 1	Make solid, face, vertex
KSFV (F, V : IN)	-1, 0, -1	Kill solid, face, vertex
MEV (V1:IN;E,V2:OUT)	1, 1, 0	Make edge, vertex
KEV (E, V:IN)	-1, -1, 0	Kill edge, vertex
MEF (V1,V2,F1:IN;E,E2:OUT)	0, 1, 1	Make edge, face
KEF (F, E:IN)	0, -1, -1	Kill edge, face
SEMV (E1:1N;E2,V,OUT)	1, 1, 0	Spilt edge, make vertex
JEKV (E1,E2:IN)	-1, -1, 0	Join edge, kill vertex

definitions can be found in [70 ].

MSFV Operator: MSFV forms the initialization operator of the set of vertices, edges and faces. It creates a new graph instance with just one node and face. MSFV is always applicable.

The corresponding inverse operator KSFV is applicable to a graph component consisting of just one face and one node. It removes the face and the node.

MEV Operator: The MEV operator subdivides the cycle of edges of a node, thereby creating a new edge and a new node. It basically acts as a "draw-an-edge" operator.

The inverse operator KEV is applicable to an edge occurring twice in a region and having no attached edges at one of its nodes. It removes one edge and one node.

MEF Operator: The MEF operator divides a face into two faces by adding a new edge between two nodes. The new edge will occur once in both new faces in reverse orientations.

The inverse operator KEF is applicable to an edge occurring in two distinct regions. By KEF, the two faces are connected by removing the edge to form a new face bounded by the remaining edges and nodes of the original regions. Note the duality between MEF and MEV.

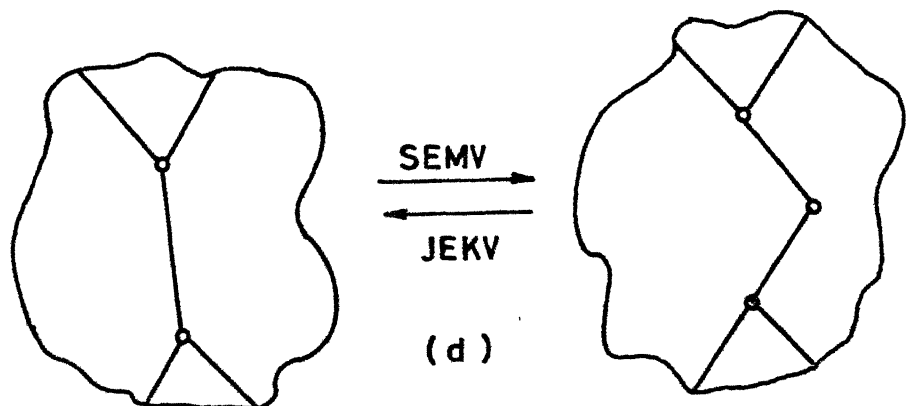
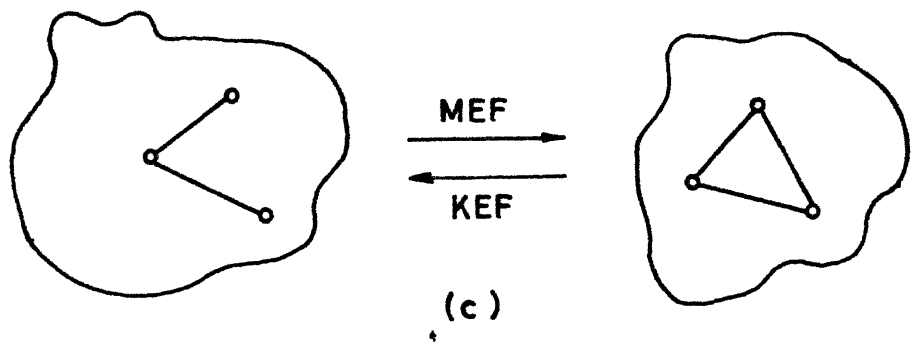
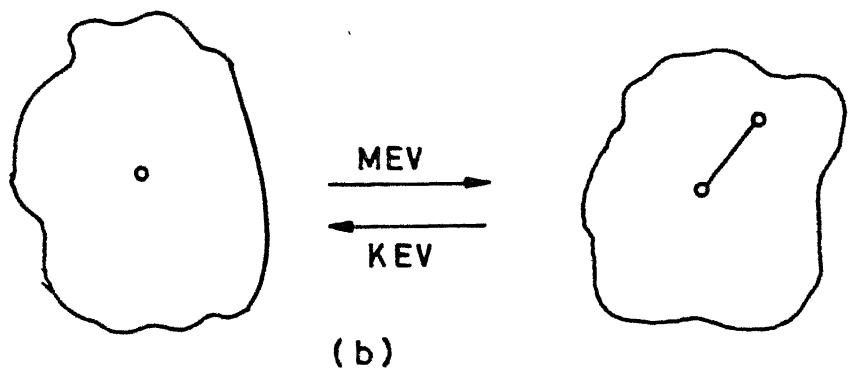
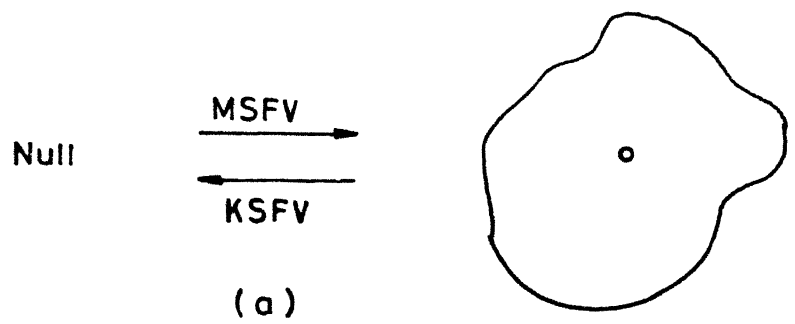


Fig.4-4 Euler operators

SEMV Operator: The SEMV operator splits an edge to introduce a new node and a new edge thereby adding a new node and a new edge to the face cycles. The new edge and node will occur in both the faces to which the original edge belongs.

The inverse operator JEKV removes an edge and a node by joining two nodes by a new edge. It changes the node and edge cycles of two faces to which the edges belong.

#### 4.3.4 Adjacency Relationships

Adjacency topology concerns the physical adjacencies of three primitive topological elements (vertices, faces and edges) on the graph. To facilitate discussion regarding this following symbols are used:

$V_i$	a vertex
$E_i$	an edge
$F_i$	a face
$\{V_i\}$	a set of vertices
$\{E_i\}$	a set of edges
$\{F_i\}$	a set of faces
$V$	total number of vertices
$E$	total number of edges
$F$	total number of faces

Since a three element topology is discussed here, there are nine distinct combination of element types in the

adjacency relationships; each is called an adjacency relationship class or topological query (TQ). These nine TQs are listed in Table 4.2. Note the duality between a vertex and a face. In particular, note that  $EE_i$  wings of an edge  $E_i$  are self dual.

A boundary data structure can be thought of as a set of adjacency relationships among the three basic topological entities [73]. Let a relation be denoted by

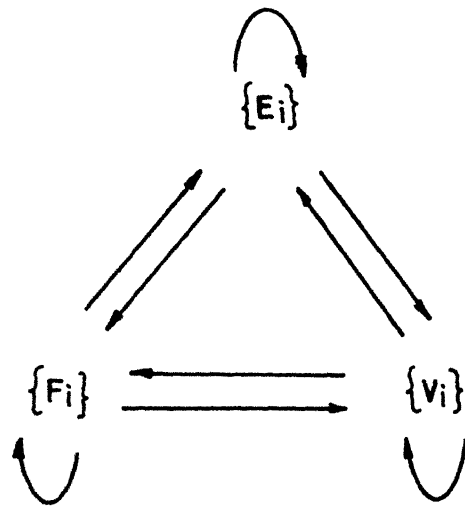
$$X \rightarrow Y \quad (4.7)$$

where  $X, Y$  can be a set of vertices  $\{V_i\}$ , edges  $\{E_i\}$ , faces  $\{F_i\}$ . A relation  $\{E_i\} \rightarrow \{V_i\}$  or simply  $E \rightarrow V$ , for example, stores two vertices for each of the edges. Hence, given an edge  $E_i$ , its associated vertices  $\{V_i\}$  can be retrieved or updated. All such nine adjacency relationships or TQs can be represented as a graph with three nodes and nine arcs, as shown in Fig. 4.5a. Figure 4.5b shows the schema for winged-edge data structure, with an edge pointing to its two vertices, two faces, and four of the possible many edges, while a face or a vertex points to one of their many edges. Note that three TQs e.g.  $E \rightarrow V$ ,  $E \rightarrow F$  and  $E \rightarrow E$  are fully stored in W-E data structure. Some TQs are stored fractionally e.g. instead of storing all the

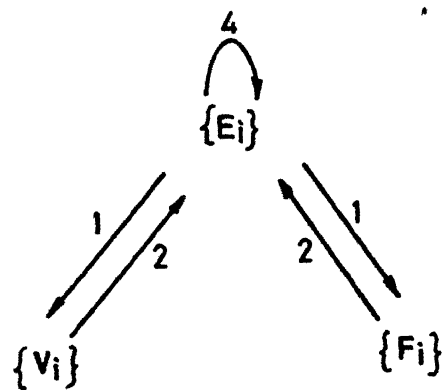
Table 4.2 : Topological Queries

	Query	Explanation
TQ1	$VV_i$	Find the clockwise-ordered circular list of vertices surrounding $V_i$ , ( $VV_i$ = variable)
TQ2	$EV_i$	Find the clockwise-ordered circular list of edges surrounding $V_i$ , ( $EV_i$ = variable)
TQ3	$FV_i$	Find the clockwise-ordered circular list of faces surrounding $V_i$ , ( $FV_i$ = variable)
TQ4	$VE_i$	Find the vertices sharing $E_i$ (ordered access), ( $VE_i$ = 2)
TQ5	$EE_i$	Find the edges adjacent to $E_i$ which share $VE_i$ vertices and $FE_i$ faces of $E_i$ , ( $EE_i$ = 4)
TQ6	$FE_i$	Find the faces adjacent to $E_i$ (ordered access), ( $FE_i$ = 2)
TQ7	$VF_i$	Find ordered circular list of vertices surrounding $F_i$ where the area of $F_i$ is found to the right when traversing the sequence of vertices, ( $VF_i$ = variable)
TQ8	$EF_i$	Find ordered circular list of edges surrounding $F_i$ , where the area of $F_i$ is found to the right when traversing the sequence of edges, ( $EF_i$ = variable)
TQ9	$FF_i$	Find the ordered circular list of faces surrounding $F_i$ where area of $F$ is found to the right when traversing the sequence of faces, ( $FF_i$ = variable)





(a) Nine relations and three entities



(b) Winged edge data structure

FIG.4.5 Schematic for edge based data structure

$EF_i$  edges for a face  $F_i$ , a face pointing to only one edge is stored. Similarly,  $V \rightarrow E$  is also stored fractionally. These two and other four TQs which are only partially stored or are not stored at all, are to be accessed by intelligent algorithms which make use of information accessibility available in the W-E data structure. Such algorithms are discussed in detail in [44].

#### 4.3.5 Storage and Time Complexity of W-E Data Structure

The storage complexity of a data structure is measured using counting formulas, using which one can assign each relation a storage cost in terms of the total number of edges  $E$  in the graph. Therefore, a set of adjacency relations with a storage cost represents a static view of the data structure. Also by the basic queries for accessing, one can describe a relation that is not stored directly in a given data structure and that can be expressed as a procedure in terms of relations that are stored. The time complexity of a data structure is measured using this set of primitive queries. Hence, here one gets a dynamic view of a given data structure by the way it is accessed indirectly or reversely.

The storage complexity of a relation  $X \rightarrow Y$  can be measured by taking the sum [45],

$$\sum_{i=1}^X YX_i \quad (4.8)$$

where X and Y can be V, E or F, and i is summed over all x. Implemented as arrays, the storage for two relations  $E \rightarrow V$  and  $E \rightarrow F$  cost  $2E$  cells each. This is because each edge  $E_i$  has two vertices  $\text{vert}[1]$  and  $\text{vert}[2]$  as well as two faces  $\text{face}[1]$  and  $\text{face}[2]$ . As there are  $E$  such edges, the total storage is  $2E$ . Hence,

$$\sum_i^E VE_i = \sum_i^E FE_i = 2E \quad (4.9)$$

As each edge has four wings, the storage cost for relation  $E \rightarrow E$  is clearly  $4E$ , i.e.,

$$\sum_i^E EE_i = 4E \quad (4.10)$$

In W-E data structure, use of fractional relation is made, i.e., instead of storing all the  $EF_i$  edges for a face  $F_i$ , a face pointing to only one edge is stored. This fractional relation  $F \rightarrow E$  gives rise to a storage cost of  $F$  as it points to only one edge, and there are  $F$  of them. Similarly,  $V \rightarrow E$  costs  $V$  in storage. The total storage is therefore  $8E + (V+F) = 8E + (E + 2) = 9E$ . [making use Euler formula, eqn. (4.6)]. Implication of this is less the number of edges in the graph, less is the

storage requirement.

The time complexity for the W-E data structure can be analyzed as follows. Since the three direct relations are  $E \rightarrow V$ ,  $E \rightarrow F$  and  $E \rightarrow E$ , the three corresponding queries TQ4, TQ5 and TQ6 can be answered in constant time  $K$  as the arrays allow direct access. For other six queries, local 'clocking' around a face or around a vertex is required, costing  $EF_i$  or  $EV_i$  in time respectively. Since  $EV_i$  is of the same order as  $EF_i$ , one can choose  $EV_i$  as the unit [72]. Hence, the total time complexity for the winged-edge is  $O(6EV_i + 3K)$ . Implication of this is that, time complexity is dependent upon a local quantity  $EV_i$  instead of a global  $E$ . In other words, processing time is independent of the size of the problem (which is equivalent to total number of edges  $E$ ), i.e. response time will not suffer as the problem size increases. Also, if the local quantity  $EV_i$  decreases (i.e. less cluster of edges around vertices, in general), access time falls linearly with  $EV_i$  making data processing time less.

#### 4.3.6 Implementation

The program here is written in FORTRAN/77 programming language. A serious drawback of using FORTRAN is that the language does not support record-structured data types, the natural way to store the winged edge data structure

information. With FORTRAN, the subscripting mechanism is chosen for providing the necessary addresses. A set of FORTRAN vectors is used to represent the various fields comprising the records in a linked list structure. A single record is represented as a cross-section of these vectors across the same subscript, and this common subscript serves as a pointer to that record. Consequently, the vector names serve as the field selectors, while the pointers are used as subscripts for various vectors.

The software support for the data structure is developed in a number of distinct layers to ensure modularity (Fig. 4.6). The lowest level contains the memory allocation and deallocation routines. Control over storage utilization is maintained by keeping all the unused records chained together into available lists. Whenever a new record is to be created, memory allocation routine remove the first available record from the availability list. When a record is deleted, it is added to the availability list so that it is reused before any new memory is employed by the data base.

The next level of software contains the routines to create and delete the elementary topological entities. These routines create and destroy vertices, edges and faces. For these processes, the record creation routines are called so that memory is allocated for the new record or memory passed onto

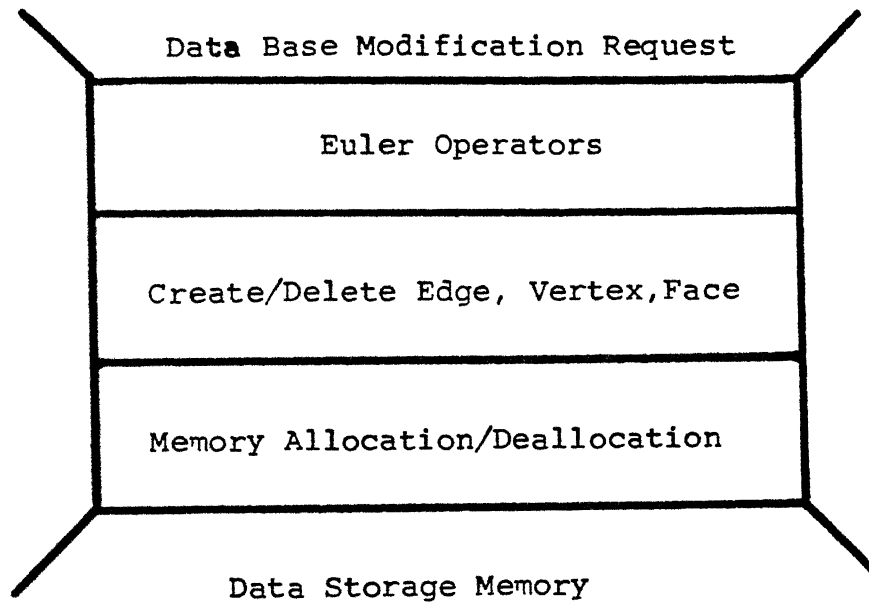


Fig. 4.6 : Layered Organisation of Data Base Modification Routines.

these record creation routines.

The software level above the routines that deal with individual topological elements are those that implement Euler operators described in section 4.3.4. These routines call the routines which create or delete edge, vertex, face to achieve various functions such as making or killing edges, vertices, faces etc., keeping topological integrity intact. Also they operate on few other fields of certain records e.g. material code field of face record.

In addition to creation and deletion routines, the data structure software also supports a number of query routines. All the query routines of Table 4.2 are supported. In addition, the following additional routines are also supported: get the list of all vertices, get the list of all faces, GET-FACE. The GET-FACE returns the face that contains a given X and Y coordinate pair. This allows the program to determine which element a user has selected with the cross-hair cursor.

A uniform approach is taken for applying such data structure to both BEM and FEM. Boundary element regions are treated<sup>as</sup> 'super' finite element type faces and they are flagged with a special face type which differentiates them from the possible adjoining finite elements. All queries and Euler operators can then be applied on such a face without any restriction or modification.

#### 4.4 Program Integration and A Sample Problem

The key aspect of the present FRACTure Propagation Code (FRAPCO) developed here is the integration of all topics discussed so far, namely, fracture mechanics, BEM and FEM, mesh generation, data base design, into one coherent program. The current version of the program operates on supermini computer ND-560/CXA system which consists of ND-500, 32 BIT main processor and ND-100, 16 BIT front end processor. The system has virtual memory capabilities which obviates the need of using secondary storage units, i.e. all the program arrays are held in 'core' and the swapping of part of any array to fast disk storage is left to the efficient system routines. The user interaction and display relies on hardware and software facilities of Tektronix 4109 (19", 640x480 pixels, 16 colours/4096 hues) and Tektronix 4107 (14", 640x480 pixels, 16 colours/64 hues) raster graphics terminals.

An important feature of FRAPCO is the modular nature of its basic program design. A common data base is maintained containing a complete current description of the structure. This data base is accessible through data access routines, which is used by all higher level routines for query into the data base or to modify the data base.



The system is menu driven. A schematic of interaction of the menu pages is indicated in Fig. 4.7. At any point of execution of the program, the user is confronted only with the options on the currently displayed menu page. This greatly helps the user who is guided by the menu page structure which is designed to help user tackle particular aspect of finite and boundary element crack propagation analysis. The finite element mesh and problem attributes change during the flow of the program either automatically or manually, under the control of the user, through graphical and/or alphanumeric key board inputs. Geometrical information can be input by using the cross-hair cursor. Previously defined elements of a data base can be identified by pointing with the cross-hair cursor. This frees the analyst from the burden of referring to nodes, elements, lines etc. by number through keyboard.

The operation of the system can best be appreciated by considering fracture propagation analysis of a practical problem. For this purpose, a rectangular plate with a crack emanating from a circular hole (Fig. 4.8) has been selected. The step by step solution of the example problem illustrates how W-E data is used to support some of the functions performed by the program FRAPCO also how the program interacts with the user for a crack propagation analysis.

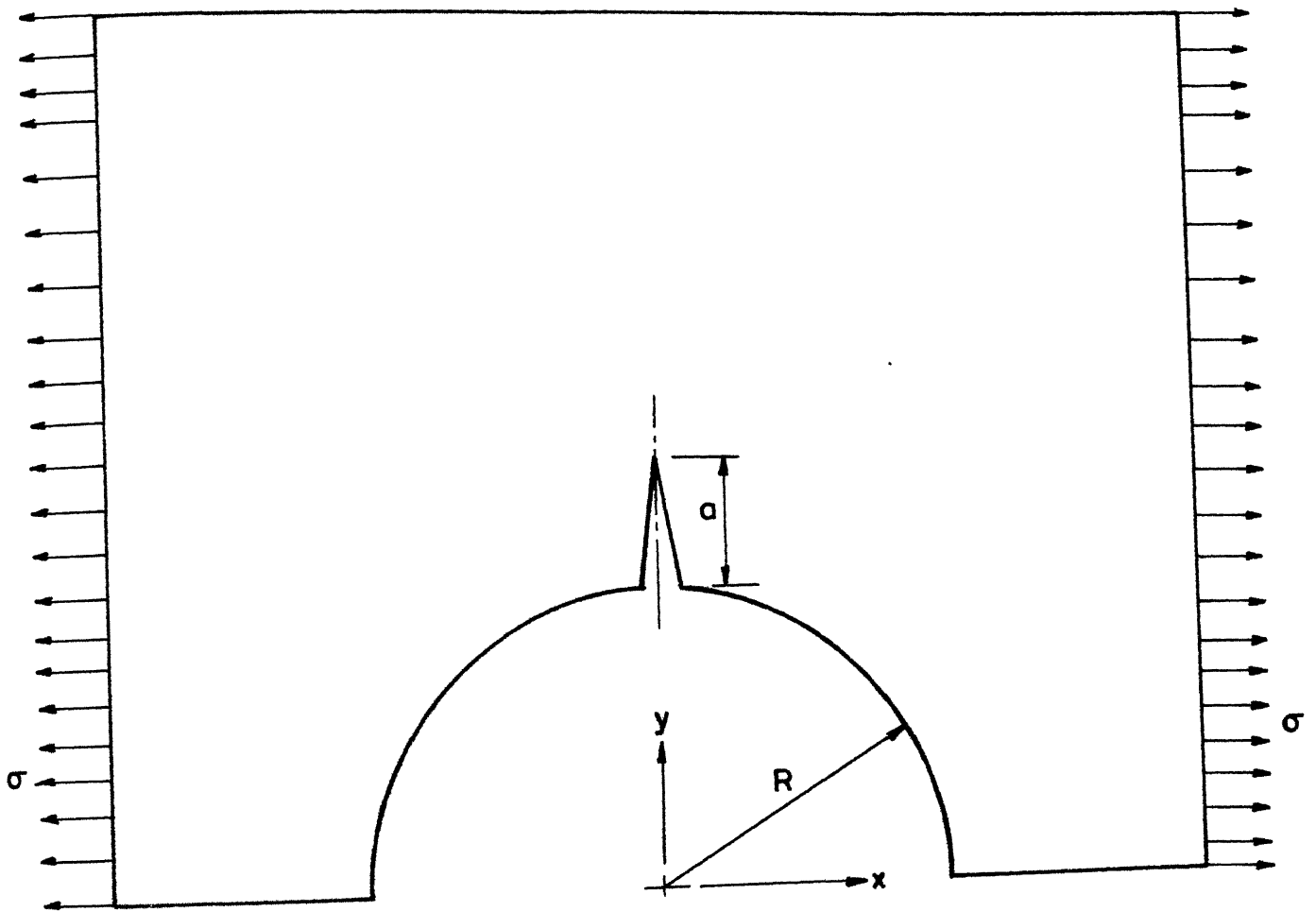


FIG. 4-8 Crack emanating from a hole in a plate subjected to uniaxial tension

#### 4.4.1 Geometry Handling

The first step in the analysis is the geometric description of the problem. Geometry handling refers to the description and modification of the geometric and topological aspect of the problem. Under this come the mesh creation and mesh modification routines. The first step in this is to specify the data limits i.e. the extent of world window which is to be transformed into device (i.e. display terminal) coordinates system for displaying the mesh.

The boundary curves defining the subregion outlines are input by the curve generating package invoked by the outline menu page, shown in Fig. 4.9. In the present program the user can select from the following curve types - straight line , circular arc or discrete. The discrete curve option is used when mathematical form of the curve is not explicitly known. Depending upon the key information (e.g. for straight line - starting and end point coordinates, number of subdivisions on the line etc.) provided by the user, the program generates and stores discrete form of the outlines. If the user is not satisfied with any particular curve, he can delete it through curve delete option by picking the curve to be deleted. Fig. 4.9 depicts the complete boundary curve description of the present problem.

The actual element mesh is generated using the display page shown in Fig. 4.10. The boundary curves are displayed

for selection by the user, who is prompted to pick the four curves defining each region. The user then chooses an element topology (3-noded triangle, 4-noded quadrilateral, 6-noded triangle, 8-noded quadrilateral, boundary elements). However, even though all such types of elements could be generated, at present, for numerical and analysis, only 8-noded quadrilateral finite element <sup>and</sup> quadratic boundary elements are implemented. The program automatically computes and displays the mesh for approval of the analyst. The domain as well as boundary nodes are created for a region which is discretized by finite elements and boundary nodes only for a region that is discretized by boundary elements. A uniform code is used to generate mesh for both finite and boundary element regions and no extra effort is required from the user to generate a coupled boundary and finite element mesh.

Display of finite element mesh is very efficient task with the winged-edge data base being maintained. It is desirable that each edge in the mesh be drawn only once. If the only information available to the program is the element connectivity table, then a check must be made for each edge to determine if it has been drawn yet. This algorithm will take  $O(n^2)$  time where  $n$  is the number of elements in the mesh. With W-E data structure, displaying the mesh is a simple <sup>task</sup>  $\angle$  Since each edge is stored uniquely, the program can traverse the linked list of edges embedded in the data structure. This process takes  $O(m)$  time where  $m$  is the number of edges in the mesh.

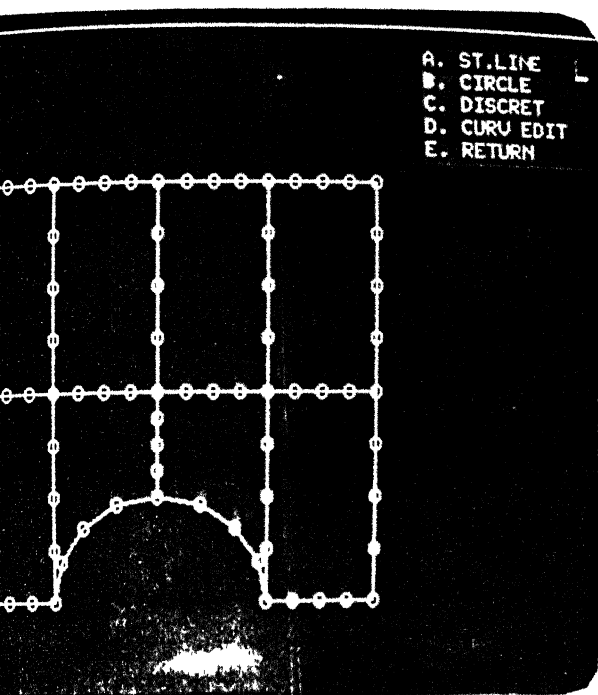


Fig.4.9 : Boundary Curve Description of the Example Problem

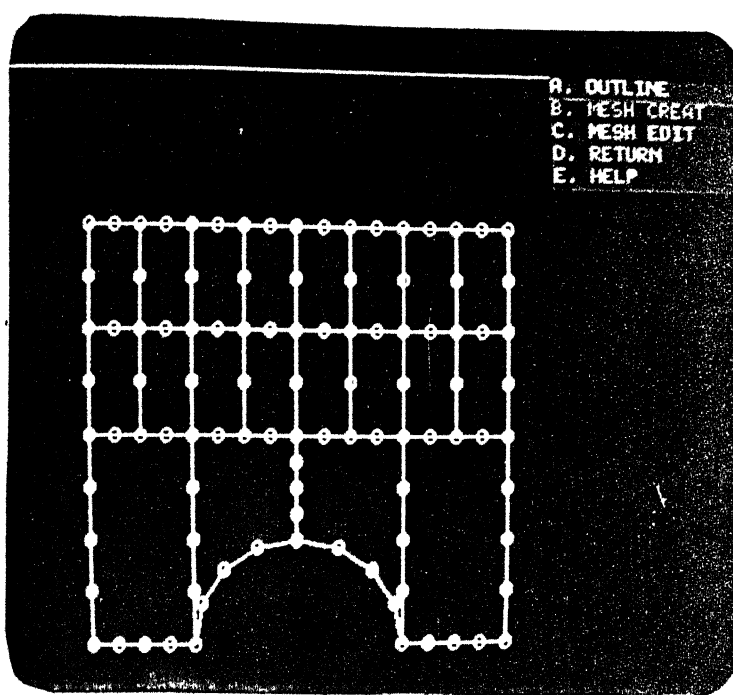


Fig.4.10 : Mesh Generation of the Example Problem

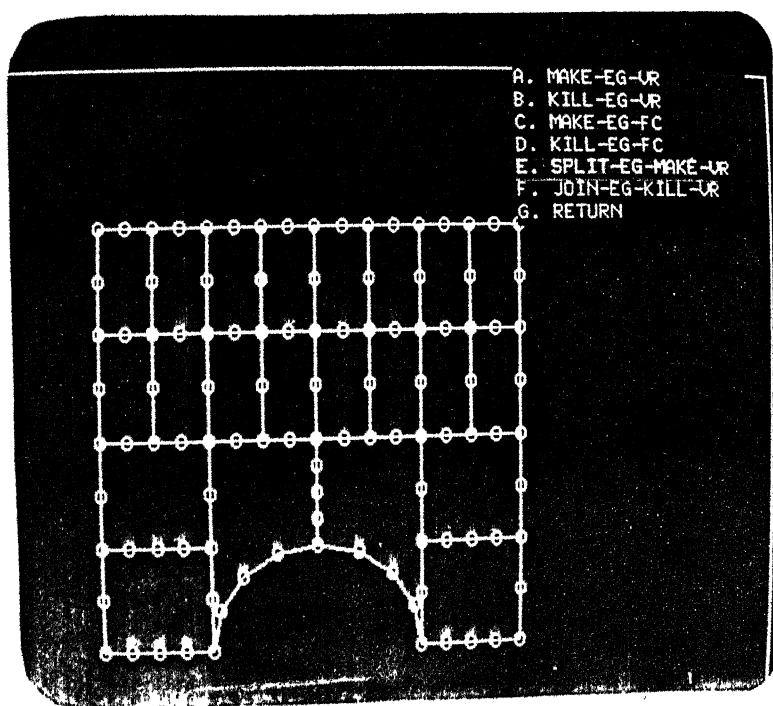


Fig.4.11 : Mesh Editing of the Example Problem

After the mesh is created, the user may wish to perform local mesh modification. This is accomplished by using the mesh editor display page shown in Fig. 4.11. All the six Euler's operators described in section 4.3.3 are provided for this purpose. The program performs the modification and query tasks through these routines to directly work on the winged edge data base. In the present example, more boundary element sub-domains are created around crack tip in a way to refine the mesh around the crack tip. The edges are created by MEV and MEF functions to arrive at the final mesh shown in Fig. 4.11.

#### 4.4.2 Attribute Editing

After the geometric description of the problem, before an analysis can be performed, additional input data such as material properties, boundary conditions, loads etc. are required. This data may be viewed as a series of attributes associated with the nodes and elements contained in the geometric description. This inputting is achieved through 'Attribute' menu page shown in Fig. 4.12 which has options for specifying both point load and distributed edge (pressure) load, displacement boundary conditions and material properties.

The process of specifying displacement boundary conditions is considerably simplified by assuming several nodes on the boundary to have same type and amount of displacement boundary conditions. Thus the user is prompted

to input the specified displacements and fixity conditions to be associated with a particular portion of the boundary. The user then indicates the portion of the boundary over which these conditions are to apply by picking the starting node and end node of the boundary portion. In the present example, the bottom edge of the plate is to have all fixed displacements in both X- and Y-direction. This 'both fixed' code is entered first through keyboard. The portion of the boundary over which this condition is to apply is then indicated by picking the right-most node (Fig. 4.12) and left-most node (Fig. 4.13) of the bottom edge of the plate. The program automatically picks all the boundary nodes along that portion of the boundary and makes all such nodes 'both fixed' type. The load boundary conditions are input similarly in turn by specifying point load and/or edge load to be applied on a group of nodes along the boundary.

If no boundary information is explicitly available to the program, this type of boundary portion identification requires an  $O(n^2)$  ( $n$  = number of elements in the mesh) search to identify boundaries. With W-E data-structure, faces in the topology that do not represent elements are flagged with a special face type. The set of edges adjacent to these faces comprise the structural boundary. Once a starting edge or node on the boundary is (graphically) identified by the analyst, the wing fields in the edge records are used to traverse an ordered list of edges and nodes along the boundary

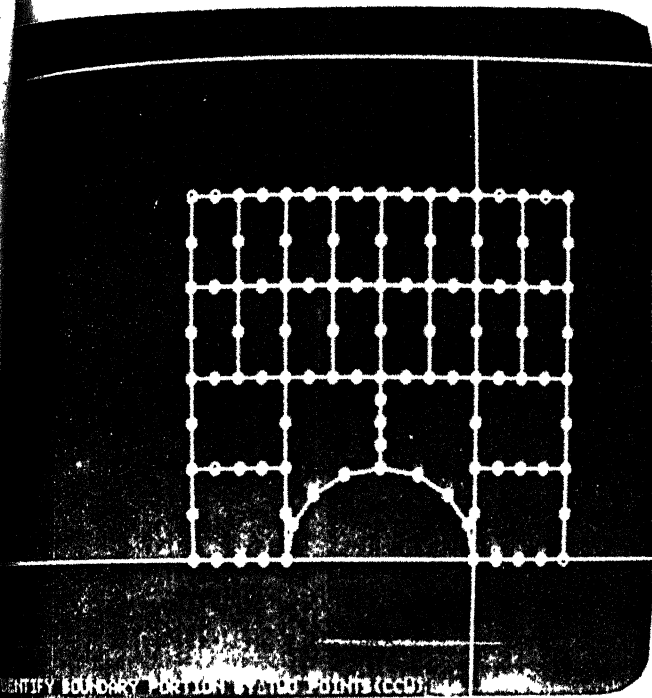


Fig.4.12 : Attribute Editing of the Example Problem

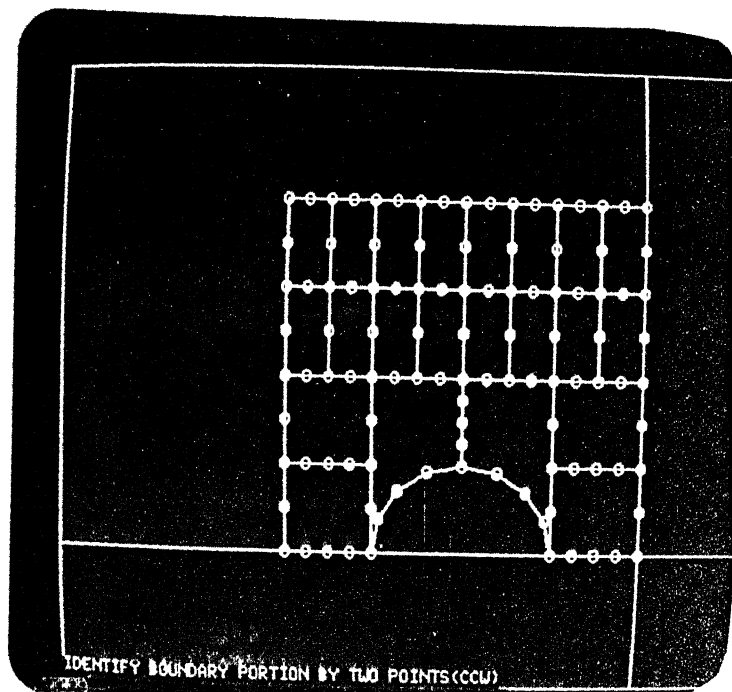


Fig.4.13 : Attribute Editing of the Example Problem

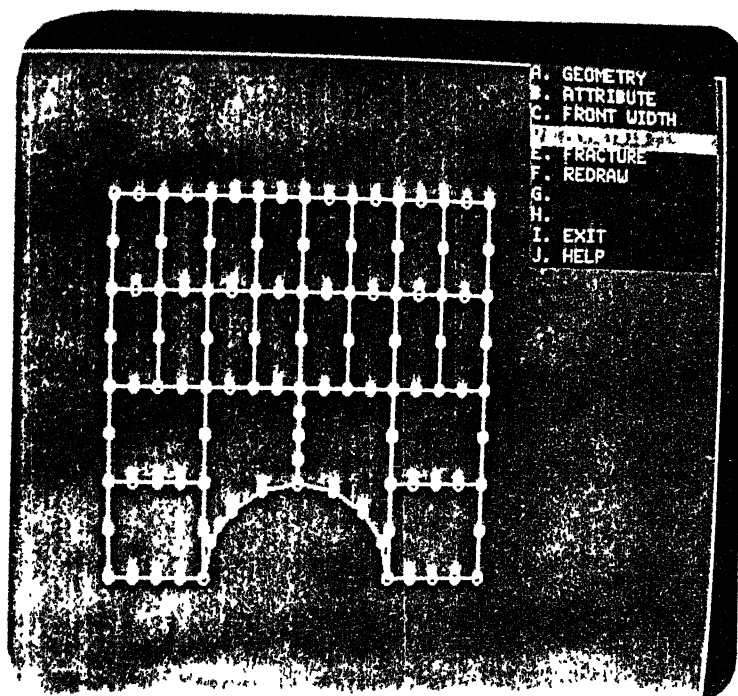


Fig.4.14 : Crack-tip Definition of the Example Problem



spanning from the indicated starting vertex or edge to the indicated stopping vertex or edge.

#### 4.4.3 Crack Tip Definition

The program FRAPCO uses quarter-point boundary elements to model the singular stress field around the crack tips. This requires modification of coordinates of all nodes adjacent to the crack tip nodes. This is achieved by the FRACTURE option in the driver menu page, as shown in Fig. 4.14.

Typically, the user picks the crack tip node by pointing to it with the mouse. The program then identifies all nodes adjacent to the crack tip node, i.e. the 'quarter-point' nodes. All the nodes adjacent to these quarter point nodes are also identified so that quarter point locations can be computed. Also some of these nodes are flagged at this stage so that their displacements can be recovered after performing the elastic analysis for calculating stress intensity factors by displacement correlation method. If only topological information available is the connectivity table, this process takes  $O(n)$  time where  $n$  is the number of elements in the mesh. The W-E data structure has a  $VV_1$  operator (described in section 4.3.4) which retrieves a list of vertices adjacent to a given vertex in  $O(m)$  time, where  $m$  is the number of adjacent vertices.

#### 4.4.4 Analysis and Remeshing

Once the geometry and attributes of the problem are specified and crack tips identified, the user initiates the finite element-boundary element analysis by invoking ANALYSIS option in the driver menu page (Fig. 4.14). The various important steps in the analysis algorithm for crack propagation are as follows:

1. Perform elastic analysis based upon the geometry of the structure, including initial crack geometry, applied displacement boundary conditions and applied loads.
2. Calculate stress-intensity factors  $K_I$  and  $K_{II}$  from elastic analysis and eqn. (3.44) or eqn. (3.45).
3. Check wheather a crack should propagate using strain energy density criterion (eqn. (2.4)) and calculate direction of propagation and amount of propagation, also by strain energy density criterion (eqn. (2.11) and eqn. (2.4) respectively), if any crack is on the verge of propagation.
4. Update geometry and remesh, if any crack propagates by extending structural boundary to reflect the propagation.
5. Return to step 1, With updated geometry and load for next step of propagation.

The key aspect in modelling discrete crack propagation and performing above analysis steps is the ability to modify the boundary element-finite element mesh automatically to reflect the current configuration. Dozens of man-hour is required to produce the results of few crack increments if remeshing is done manually [ 71 ]. The program FRAPCO has the ability to perform automatic local remeshing around the crack tip for each crack increment. The remeshing process is performed in following steps:

1. From the initial direction of crack axis and the predicted angle of crack extension with respect to crack direction determine the angle of propagation in global coordinates.
  2. Replace the quarter-point nodes to their initial mid-side position, to remove the local singularity.
  3. Modify the structural boundary to reflect the crack extension.
  4. Create a transition mesh that joins the new location of crack tip to the unmodified portion of the boundary.
  5. Adjust the mid-side nodes of the new elements around crack tip to quarter point location and flag these and quarter-points from which stress intensity factors will be evaluated in the next step of analysis.
  6. Display the modified mesh, to allow the user to interactively perform final adjustments.
-

An implication of the 3rd step in remeshing, i.e. modifying structural boundary, is that when the program FRAPCO modifies a certain portion of the mesh, structural boundary information must be retained. Thus, before an edge is deleted, a check must be made to see if the edge makes up a portion of the structural boundary. Since each edge is adjacent to two faces, so if one of these faces is flagged with the face type indicating that it is not an element, the edge is not deleted. The extension of structural boundary is then accomplished by the Euler operator SEMV (described in section 4.3.3). An existing edge which makes up structural boundary is split to extend the boundary to describe the new crack configuration.

The fourth step, i.e. creating a transition mesh to join quarter-point elements with the unmodified portion of the mesh is achieved by modifying the vertices definitions of edges around the crack tip. Edges adjacent to the new and old locations of the crack tip are redrawn and the modified mesh is displayed for the user to perform minor mesh editing before going into next step of analysis.

In each cycle of analysis, before updating the geometry, the user is required to choose between the option of updating geometry of all the elements (by adding displacements to all nodal point coordinates and local remeshing to reflect crack extension) or updating geometry of two boundary element regions around the crack tip (by local remeshing to reflect the crack extension only). In case of

'Stiff' material, the finite elements away from the crack tip do not deform considerably and thus fairly accurate results could be obtained just by updating geometry and recalculating the two boundary element stiffness matrices only. This amounts in substantial saving in cost of analysis as finite element stiffness matrices need not be recalculated.

The stress intensity factors are displayed at the end of second-step of each analysis cycle. User can use any criterion to calculate amount and angle of propagation or use the built-in strain energy density function to get the same. This type of interaction of FRAPCO itself can aid in the technique and theory development cycle. Because the program can at any time display graphically its current state and prompt the user for input of certain important parameters (such as angle and amount of propagation), the fracture propagation algorithms need not initially be developed in a complete form. Operations which will ultimately be performed under algorithmic control can be performed or guided by the user manually. This ability not only speeds the technique development cycle, it also often adds insight into how control algorithms should be constructed. In this way a new theory can be tested and accepted or rejected.

The coupling strategy is employed in the program FRAPCO in such a way that it can perform numerical analysis by BEM alone or by FEM alone or by a coupled BEM-FEM in a

unified manner. This kind of integrated fracture mechanics analysis system is very useful in the sense that both BEM and FEM are available in a single code, so that the user can use any one of them at his own discretion. The user can start with any kind of mesh and interactively change it to put either a few finite elements or a few boundary elements around the crack tip at any point during the analysis. Thus debate regarding which method is better for a particular problem becomes superfluous.

## CHAPTER 5

### RESULTS AND DISCUSSIONS

#### 5.1 Introduction

In this chapter results of analyses performed with the program FRAPCO are presented. The accuracy and convergence of solution obtained by symmetric boundary element formulation in fracture mechanics problem is studied with refining discretization in section 5.2. A comparison of time and storage requirements of coupled BEM-FEM and FEM is presented next in section 5.3. After getting a feeling about convergence characteristics, time and storage requirements the program FRAPCO is tested with a few test problems representing different kinds of fracture mechanics problems in section 5.4. Having checked the working and accuracy of the program, a stable crack propagation study of a centre crack in a rectangular plate is undertaken in section 5.5. The effect of initial crack size on various parameters such as propagation initiation load, critical load for bringing the crack on verge of global instability etc. is also presented in section 5.5. Finally, section 5.6 brings out the results of mixed mode crack propagation of an angled notch.

## 5.2 Convergence of Symmetric BEM Formulation

The success of a coupled BEM-FEM analysis depends on the accuracy of symmetric boundary element formulation. In order to assess the accuracy obtainable by the symmetric coupling theory, a numerical convergence study is undertaken with the aid of a mixed-mode centre slant crack-problems, as shown in Fig. 5.1. The accuracy of stress intensity factors  $K_I$  and  $K_{II}$  are studied since their correct determination ensures correctness of calculation of the amount of propagation and angle of propagation determined through equation (2.4) and equation (2.11) respectively and also because they reflect the correctness of the stress-analysis near the crack tip.

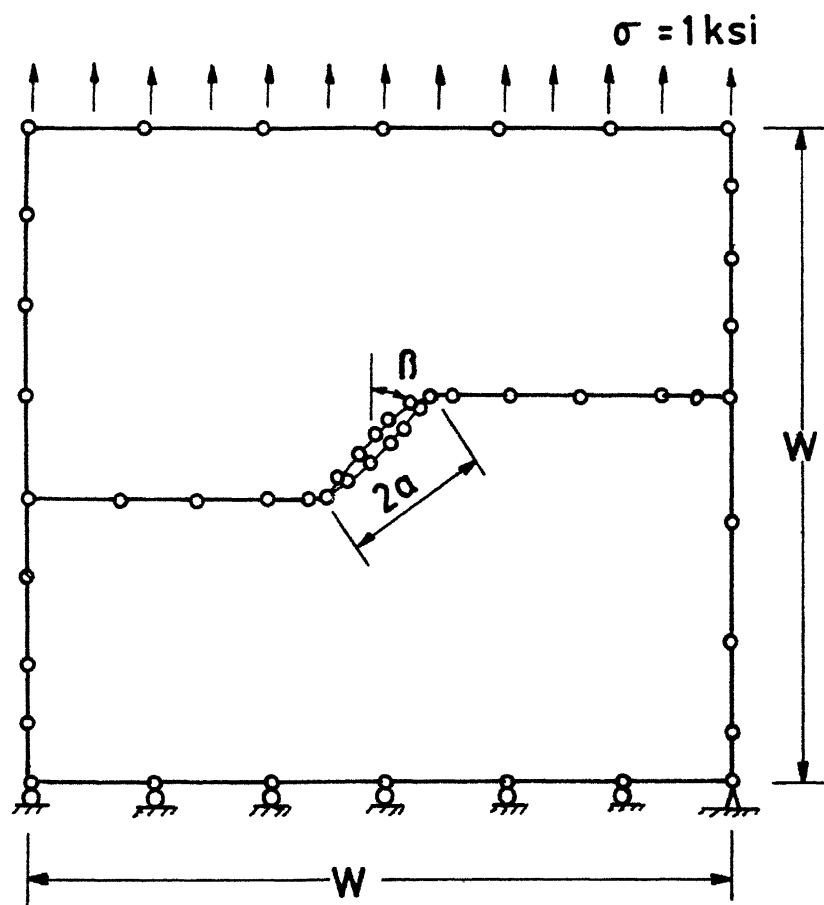
The problem is analysed with fixed size singular elements ( $l/a = 0.10$  where  $l$  = length of crack tip boundary element and  $a$  = half crack length) at the two crack tips and increasing number of non-singular quadratic boundary elements over the rest of domain. The problem is representative of a crack in an infinite domain. The results are compared with the exact results of a crack in an infinite plane [75] :

$$K_I = \sigma \sqrt{\pi a} \sin^2 \beta$$

$$K_{II} = \sigma \sqrt{\pi a} \cos \beta \sin \beta$$

in which  $\beta$  is the angle between the vertical load axis

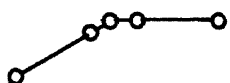




$$E = 5250 \text{ ksi}, \nu = 0.2, W = 10'',$$

$$2a = \sqrt{2} \text{ inch}, \beta = 45^\circ$$

Crack tip element



The diagram shows a small section of the crack tip element, consisting of a series of nodes connected by lines, forming a small, elongated shape.

FIG. 5-1 Typical BEM mesh for centre slant crack problem

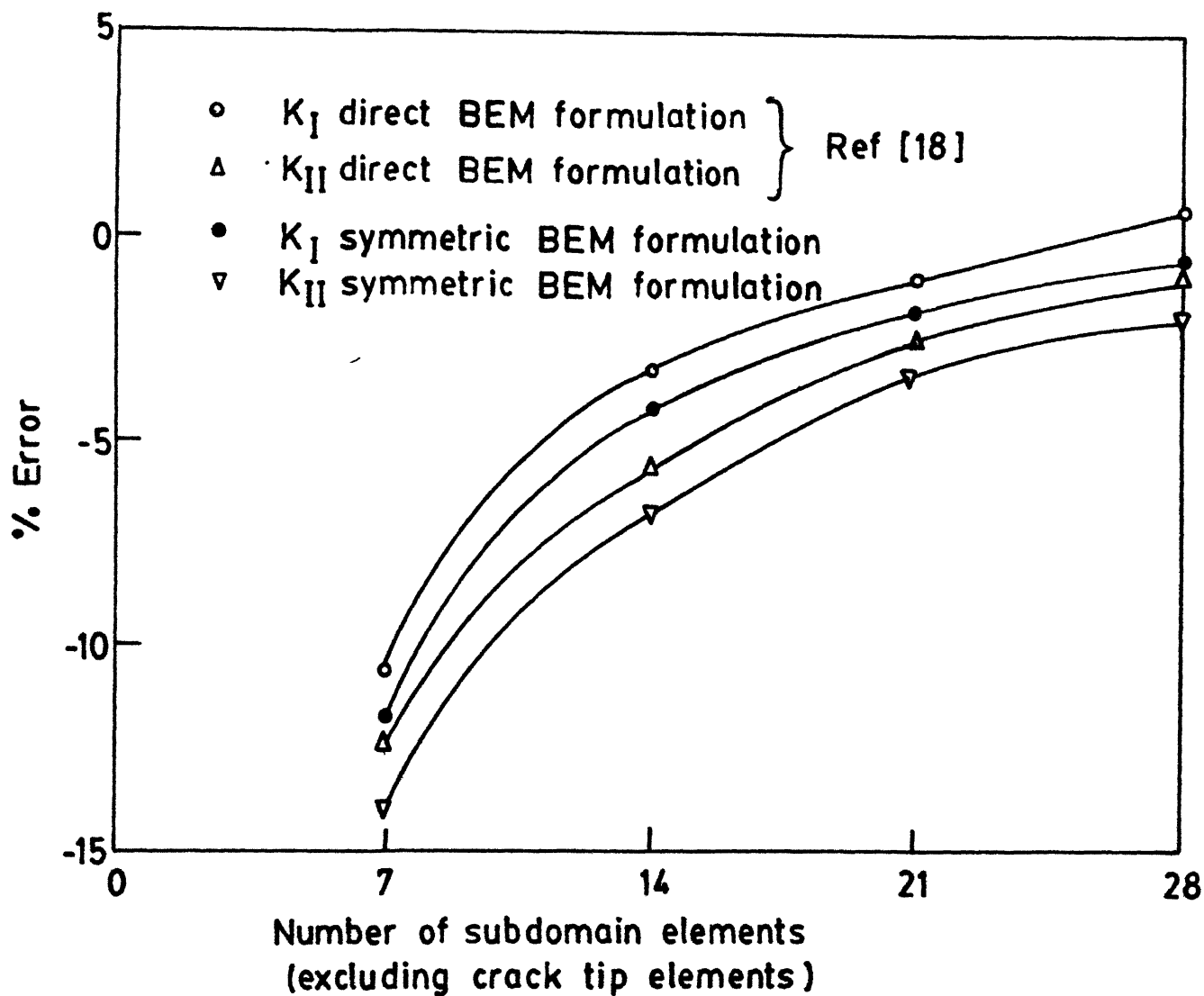


FIG. 5.2 Percent error in the computed SIF for slant crack problem

and the crack axis as shown in Fig. 5.1 . No finite length and width correction factors are applied to the SIFs of infinite plane problem as these correction factors would be quite small [75].

Results of the convergence study is presented in Fig. 5.2, in which results of symmetric formulation are plotted along with the results of conventional BEM [18] with same number of subdomain elements. From Fig. 5.2, it is clear that a better rate of convergence is obtained by conventional BEM as indicated by the steeper slopes of curves. But even though slower rate of convergence is achieved by the symmetric BEM formulation, a reasonable number of subdomain elements produce results comparable to that obtained by conventional BEM.

### 5.3 A Comparison of Time and Storage Requirements of Coupled BEM-FEM and FEM

For comparing the coupled BEM-FEM with FEM from time and storage requirements view point, a centre crack problem as shown in Fig. 5.3 has been chosen. The discretization for the coupled method is obtained by removing few finite elements from the crack tip creating a boundary element region there. For comparison, the number of nodes on boundary is kept same for both FEM and coupled method. Also, in both the cases, singular quarter point elements are used around the crack tip, as

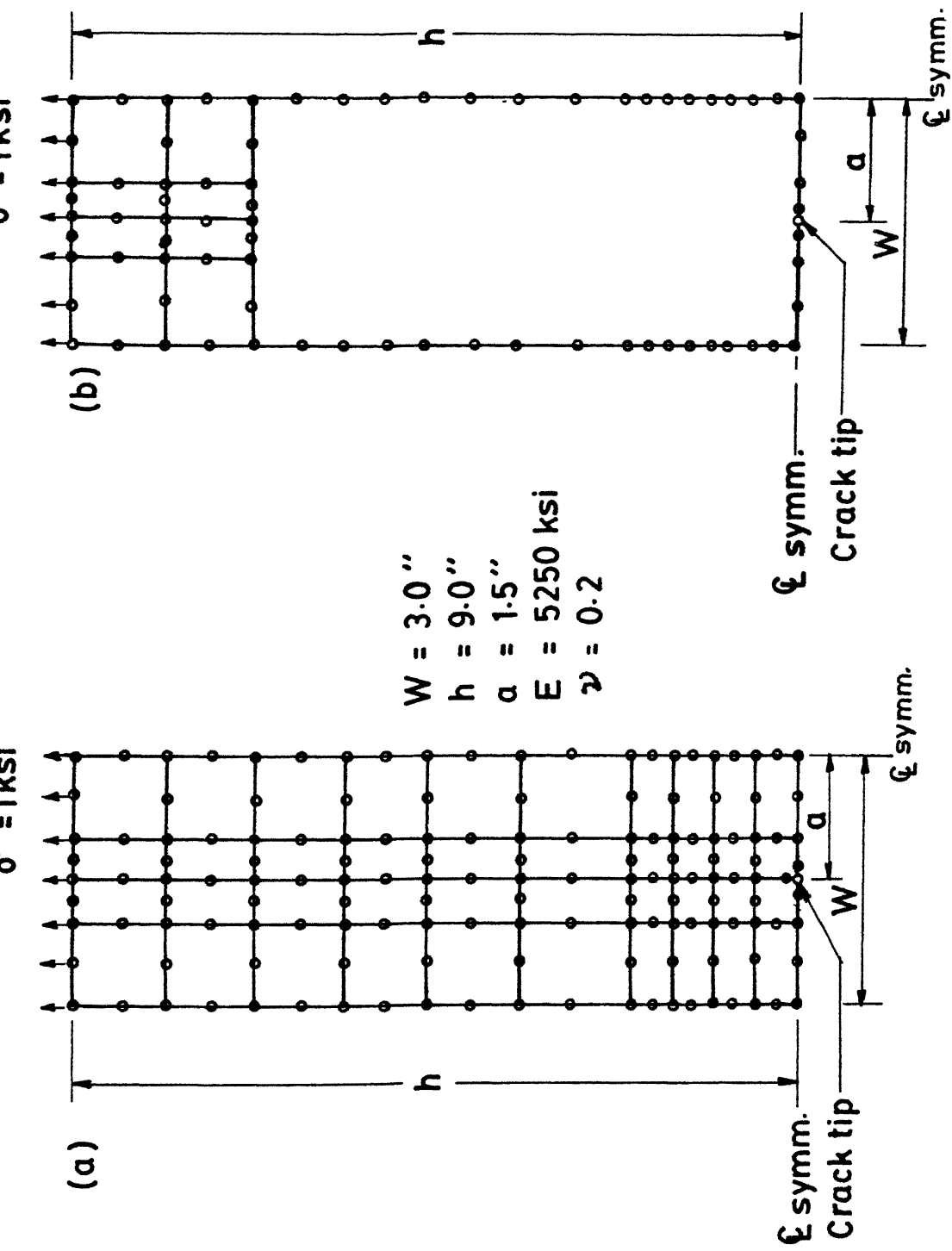


FIG. 5.3 Centre crack problem (a) Finite element mesh (b) Coupled BEM-FEM mesh

shown in Fig. 5.3. Similar interactive graphics preprocessors are used to input both the meshes and all computations are done on same ND-560/CXA machine.

Table 5.1 summarizes the salient characteristics of time requirements for FEM and coupled method. As observed from table 5.1, coupled method performs better than FEM in terms of CPU times with comparable accuracy of results being obtained. The equation formulation time is less because less number of nodes and elements are used in coupled analysis for which less computation and less communications between secondary disc storage units and core memory are required. Equation solution time is also less because of resulting smaller system of equations and also because the Frontal solver has to address the secondary disc storage units less number of times. For the coupled method, the preprocessing and other related operations, called data-handling time in Table 5.1, shows marked improvement over FEM in time requirements. This is because less number of edges, faces and vertices are used in coupled boundary element-finite element mesh resulting in a substantial reduction in mesh display, mesh editing, attribute editing and remeshing timings. As pointed out in section 4.3.5, the time complexity of topological queries (TQ) of W-E data structure is  $O(6EV_i + 3K)$  where  $K$  is constant access time and  $EV_i$  denotes the linear

TABLE 5.1 : Centre Crack Problem: Comparison of  
Typical FEM and Coupled BEM-FEM  
Solution Times

	FEM	Coupled BEM-FEM
Preprocessing time (man-minute)	6	6
Number of Elements	40	9 <sup>#</sup>
Number of Nodes	149	76
Equation formulation time (CPU-sec)*	243	132
Equation solution time (CPU-sec) *	135	60
Data handling time (CPU-sec) *	51	29
Total solution time (CPU-sec) *	469	221
Mode I stress intensity factor **	1.190	1.189

(\*) ND-100+ND-500 CPU-Time

(\*\*) Ref. [83] - 1.187.

(#) Number of boundary elements in boundary  
element region - 24

dependence of access time on the local quantity  $EV_i$ , the number of edges around vertex  $V_i$ . As the quantity  $EV_i$  is much less in coupled boundary element-finite element mesh than finite element mesh, so in general, access time for topological queries are less in coupled method. As pointed out in section 4.4, the TQs are used hundreds of time during mesh generation, mesh editing, remeshing, attribute editing, preparation of data compatible to numerical analysis subroutines (e.g. the  $VF_i$  query is used for each element to extract the node-element connectivity from W-E data base during stiffness matrix calculation). This advantage would be more prominent in analysis of large system where thousands of edges, faces and nodes are used, making the data handling time comparable to the numerical solution time.

In this connection it is interesting to note the time and algorithmic complexity involved in the remeshing to reflect configuration change due to crack propagation. In program FRAPCO boundary elements are used around the crack tip which makes the algorithm of remeshing (described in section 4.4) considerably simpler than remeshing algorithms of FEM, e.g. as described in [47]. As only three quarter point nodes are used around the crack tip node, so the search made by  $VV_i$  query to locate such nodes around the crack tip takes much less time than corresponding time for FEM remeshing. In [47], a remeshing

strategy is described in which a number of  $VF_i$  and  $EF_i$  queries are made which is not required in the remeshing algorithm used by FRAPCO, making amount of overall remeshing time for each crack much smaller.

A comparison of the storage requirement for the full topological data structure of the meshes of Fig. 5.3 is shown in Table 5.2. Assumptions made to allow comparison are that all the fields of each record structure are 32 bits in length and that an eight-bit byte is the minimum-size storage unit. Sizes of face (40), vertex (24) and edge (40) records were derived from the support record structure defined in section 4.3.8. All the fields of each record structure were considered, though some of the fields are implementation dependent, e.g., the maximum and minimum values of X- and Y-coordinate values of a face could be dropped in a storage and time trade-off. These choices have a tendency to minimize the apparent difference between FEM and coupled BEM-FEM storage requirements; the figures presented are therefore sort of best-case difference. An implementation which tries to reduce storage requirements by dropping some of the fields (which could be accessed through some search algorithms thereby increasing the time complexity) would increase the percentage of space attributed to differences in the FEM and coupled BEM-FEM meshes.



TABLE 5.2 : Typical Storage Requirements for FEM and Coupled FEM-BEM Mesh

Mesh Type	Number of topological entities			Total size in Bytes	% increase of total-size
	Face	Edge	Verts		
Coupled BEM-FEM	9	84	76	5544	56.33
FEM	40	188	149	12695	

For the type of mesh considered, finite elements mesh requires about 56% more storage than coupled BEM-FEM mesh. This is to be expected as finite element mesh consist of more number of edges, faces and vertices. As pointed out in section 4.3.5, considering only the pointers absolutely necessary, the W-E data structure storage requirement is  $9E$  where  $E$  is the total number of edges in the graph. This means that storage requirements increases linearly with  $E$ . As FEM mesh has a higher edge to face ratio than the coupled BEM-FEM mesh, the storage requirement for FEM shows dramatic increase. This factor is of utmost importance, particularly in case of very large structure where thousands of edges, faces, vertices are used and it tilts the balance in favour of coupled analysis.

#### 5.4 Testing of the Program

Having obtained a fair idea about the number of elements to be used to get acceptable results from the convergence study presented in section 5.2 and associated storage and time complexity requirements from section 5.3, testing of the program FRAPCO is undertaken in this section. Three fracture mechanics problems are studied in which stress intensity factors of a given cracked body are determined and compared with analytical and/or numerical

results reported in literature. Again the stress intensity factors are taken for the purpose of comparison as they give fair estimates of accuracy of the solution obtained by FRAPCO, as pointed out in section 5.1. The material values used for the problems in this section are  $E = 5250 \text{ ksi}$  and  $\nu = 0.20$ .

#### TEST PROBLEM 1

The coupled BEM-FEM is applied to solve a centre crack problem. A typical mesh shown in Fig. 5.4. The problem is doubly symmetric about the crack axis and hence only one quarter of the plate is considered for analysis. The mode-I stress-intensity factor is computed for a fixed height to width ratio ( $h/b$ ) of 2.0 but varying half crack length to width ratio ( $a/b$ ). The results are presented in Table 5.3 and they are compared with that of Isida [83].

#### TEST PROBLEM 2

The branch crack problem of Fig. 5.5 is studied next. This kind of branching occurs during many situations in real life such as, mixed mode loading, dynamic crack propagation etc. The problem, shown in Fig. 5.5, is composed of two branches of equal length in a square plate under uniform tension. A very large value of  $W/a$  ( $= 50$ ) is taken and in Table 5.4, stress-intensity factors at tip A are compared with that of Kitagawa et.al. [80] who solved the same problem for an infinite plate.

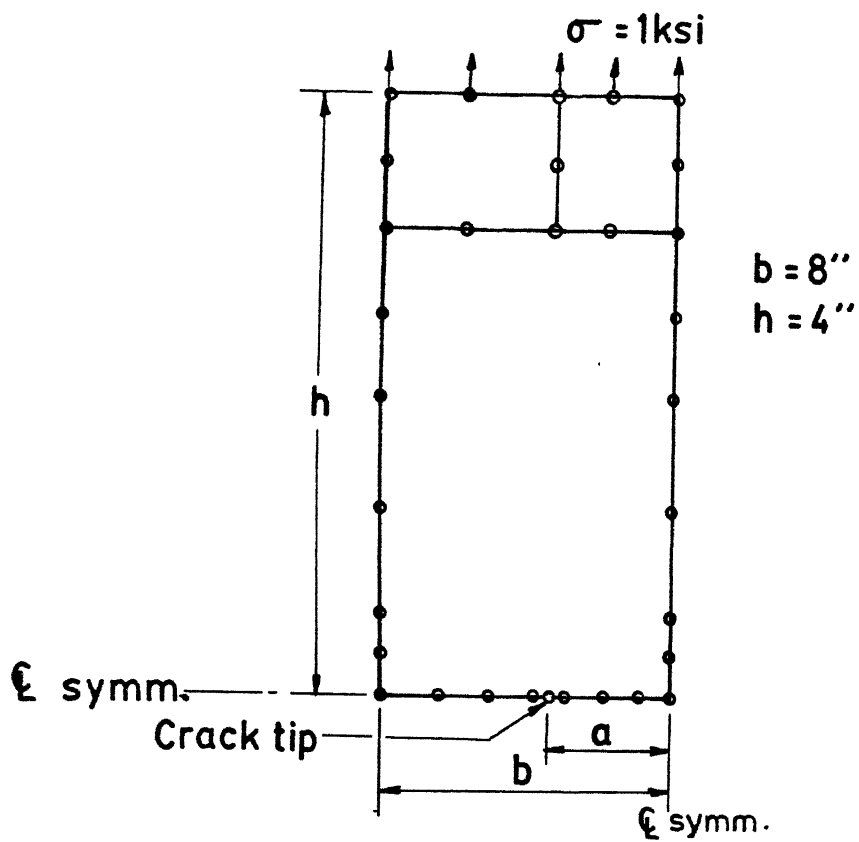


FIG. 5-4 Centre crack problem

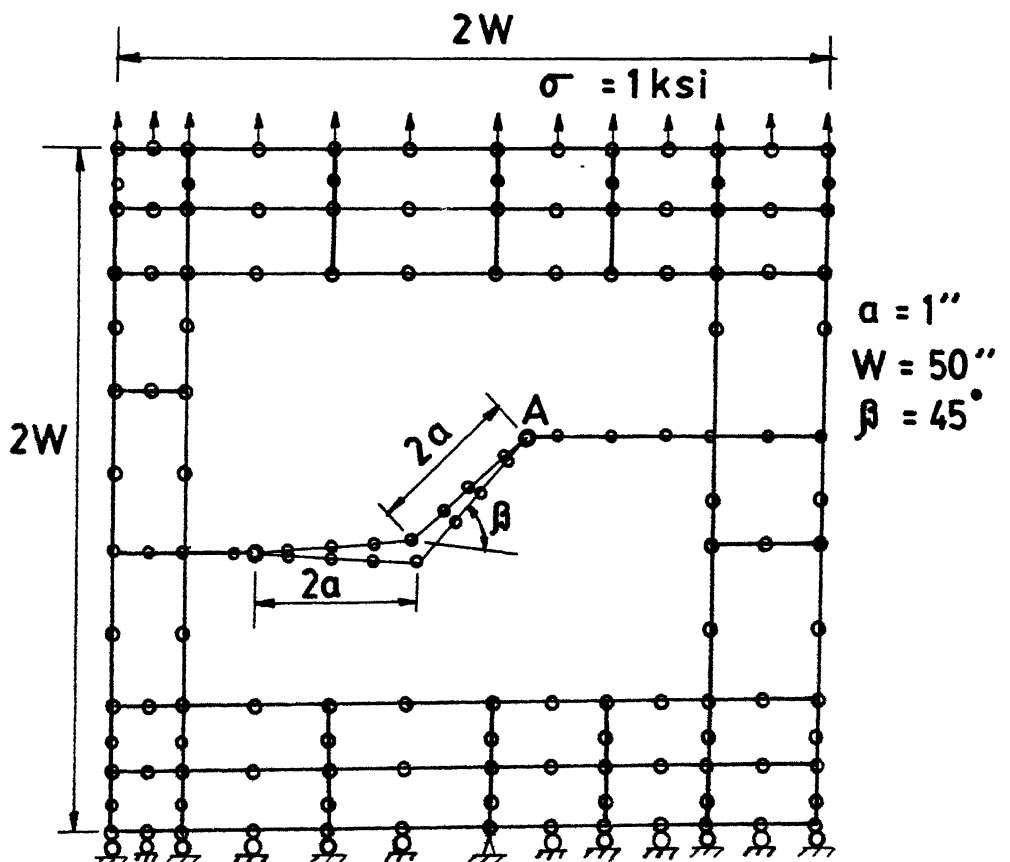


FIG. 5-5 Branched crack problem

TABLE 5.3 : Stress Intensity Factor for a Centre Crack: Test Problem-1

a/b	$K_I/\sigma \sqrt{\pi a}$ -present analysis	$K_I/\sigma \sqrt{\pi a}$ -Ref [ 83 ]
0.3	1.068	1.058
0.4	1.126	1.109
0.5	1.208	1.187
0.6	1.314	1.303

TABLE 5.4 : Stress Intensity Factor for a Bent Crack : Test Problem-2

$K_{IA}/\sigma \sqrt{2\pi a(1+\cos\beta)}$		$K_{IIA}/\sigma \sqrt{2\pi a(1+\cos\beta)}$	
Present analysis (W/a = 50)	Ref.[ 80 ] (W/a = $\infty$ )	Present analysis (W/a=50)	Ref [ 80 ] (W/a = $\infty$ )
0.574	0.596	0.654	0.641

TABLE 5.5 : Stress Intensity Factors for a Crack  
Emanating from a Circular Hole:  
Test Problem-3

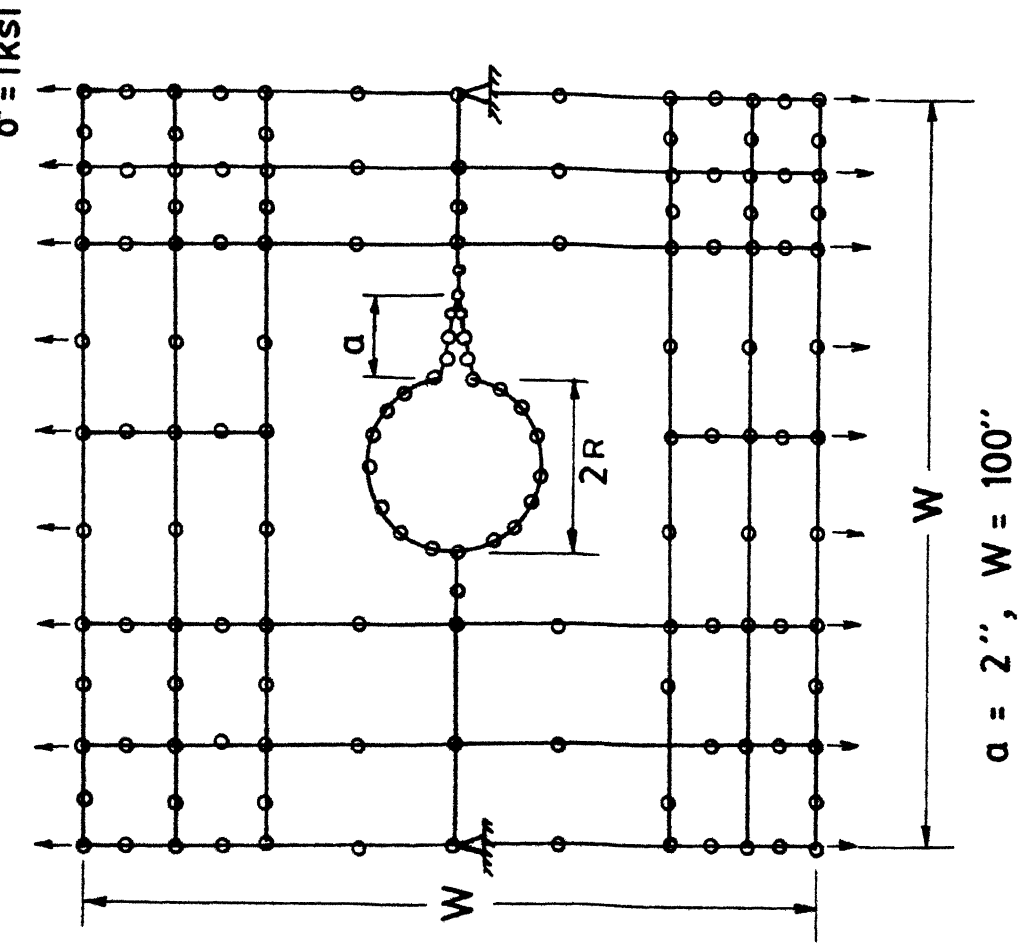
a/R	$K_I/\sigma\sqrt{2\pi}$	$K_I/\sigma\sqrt{2\pi}$
	Present analysis (W/a=50)	Ref. [ 81 ] (W/a=∞)
0.2	2.36	2.30
0.3	2.11	2.04
0.4	1.93	1.86
0.5	1.85	1.73

### TEST PROBLEM 3

A crack emanating from a circular hole in a plate subjected to uniaxial tension (Fig. 5.6) is solved next by the present program FRAPCO. The problem represents a crack in a region of high stress concentration. A typical mesh to perform the analyses is shown in Fig. 5.6. Again a very large value of  $W/a$  ( $= 50$ ) is taken and results are compared with Bowie's results [81.], who solved the problem for an infinite plate. Table 5.5 brings out the comparison of mode I stress intensity factor values of crack tip for various hole radius.

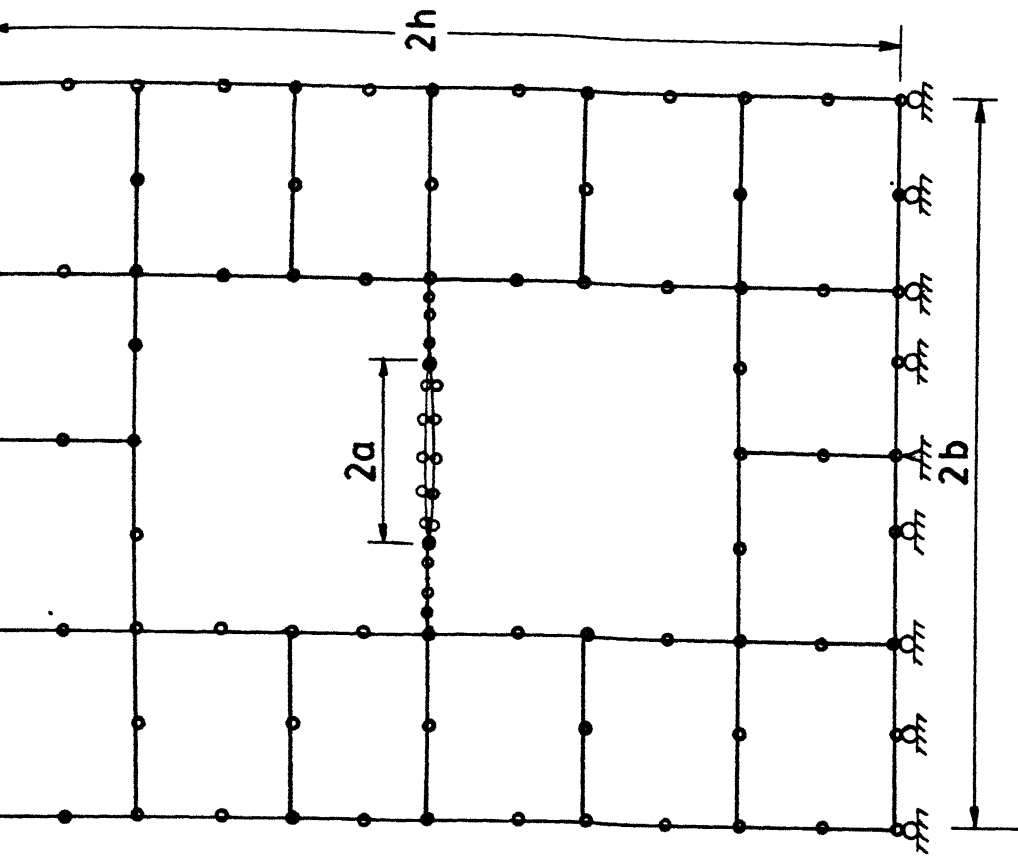
### 5.5 Propagation of a Centre Crack

The process of slow stable growth of a centre crack in a rectangular sheet is studied using the present program FRAPCO. A plate of height  $2h$  and width  $2b$  with a centre crack of length  $2a$  (Fig. 5.7) is considered. The plate is subjected to uniform uniaxial tensile stress perpendicular to crack plane. The influence of crack's geometry on the process of stable crack growth is also addressed and results are compared with that of Gdouto [79]. The width  $2b$  and height  $2h$  of the panel are taken to be 20 inch and 16 inch respectively, while the crack length is varied.



$a = 2''$ ,  $W = 100''$

FIG. 5.6 Crack emanating from a hole



$h = 8''$ ,  $b = 10''$ ,  $E = 3 \times 10^7$  psi,  $\nu = 0.3$ ,  
 $r_0 = 1.2 \times 10^{-3}$  inch,  $S_c = 7.7 \times 10$  lb/in,  
 $(dw/dv)_{cr} = 26,684$  in-lb / in<sup>3</sup>

FIG. 5.7 Center crack problem



The crack propagation initiation load,  $\sigma_i$  is calculated using strain-energy density criterion for various initial crack lengths. Fig. 5.8 presents variation of  $\sigma_i$  versus the initial crack length. As seen from the Fig. 5.8,  $\sigma_i$  decreases with initial crack length. Thus longer cracks require less initiation load as stress intensity factors increase rapidly with the crack length.

After the crack initiation, the crack grows in a stable manner until global instability is reached. In order to study the process of stable crack growth, the applied stress  $\sigma$  is increased by constant interval and the corresponding crack increments  $\Delta r$  are calculated by strain-energy density criterion, as explained in section 2.3. The process is continued until the crack extension size  $\Delta r$  becomes equal to critical ligament size  $r_c$ , which corresponds to global instability. Table 5.6 presents the successive crack increments  $\Delta r$  of a growing crack of length  $a = 2$  inch, as stress  $\sigma$  is increased at a constant interval of  $\Delta\sigma = 2.0$  ksi. The first value of the applied stress in the table  $\sigma_i = 38.7$  ksi corresponds to the critical stress for crack initiation, and the last value  $\sigma_c = 64.7$  ksi corresponds to the critical stress for global instability. The corresponding results of Ref. [79] are :  $\sigma_i = 39.8$  ksi and  $\sigma_c = 58.2$  ksi. Though crack propagation initiation load  $\sigma_i$  corresponds well with that of Ref. [79], the critical stress corresponding to global instability  $\sigma_c$  shows significant deviation from that in Ref. [79]. This is expected because in Ref. [79], load

TABLE 5.6 : Values of Relevant Parameters for Various Crack Steps of a Growing Centre Crack

Number of crack increment	Applied stress $\sigma$ (KSi)	Crack increment $r(\text{in}) \times 10^{-2}$	Half crack length $a(\text{in})$
1	38.7	1.0816	2.001081
2	40.7	1.1937	2.002275
3	42.7	1.3116	2.003586
4	44.7	1.4351	2.005021
5	46.7	1.5644	2.006585
6	48.7	1.6996	2.008285
7	50.7	1.8405	2.010125
8	52.7	1.9874	2.012113
9	54.7	2.1403	2.014253
10	56.7	2.2991	2.016552
11	58.7	2.4641	2.019017
12	60.7	2.6352	2.021652
13	62.7	2.8126	2.024465
14	64.7	2.9963	2.027461

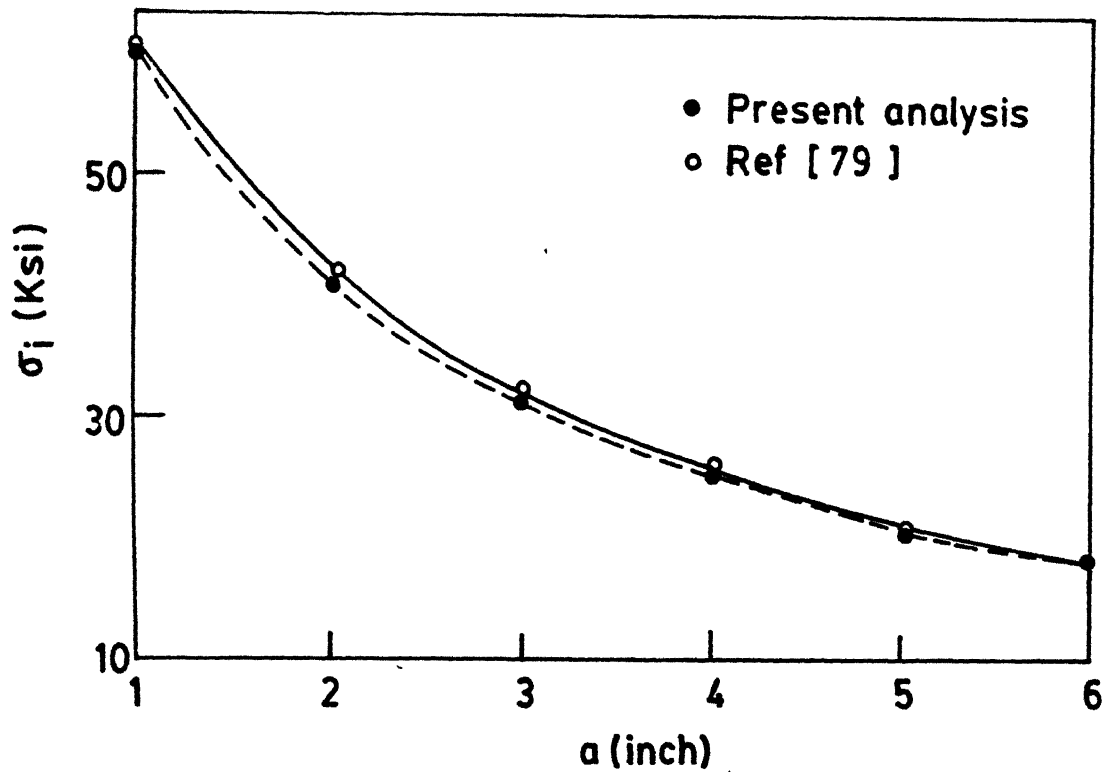


FIG.5-8 Critical stress for crack initiation vs. initial half crack length

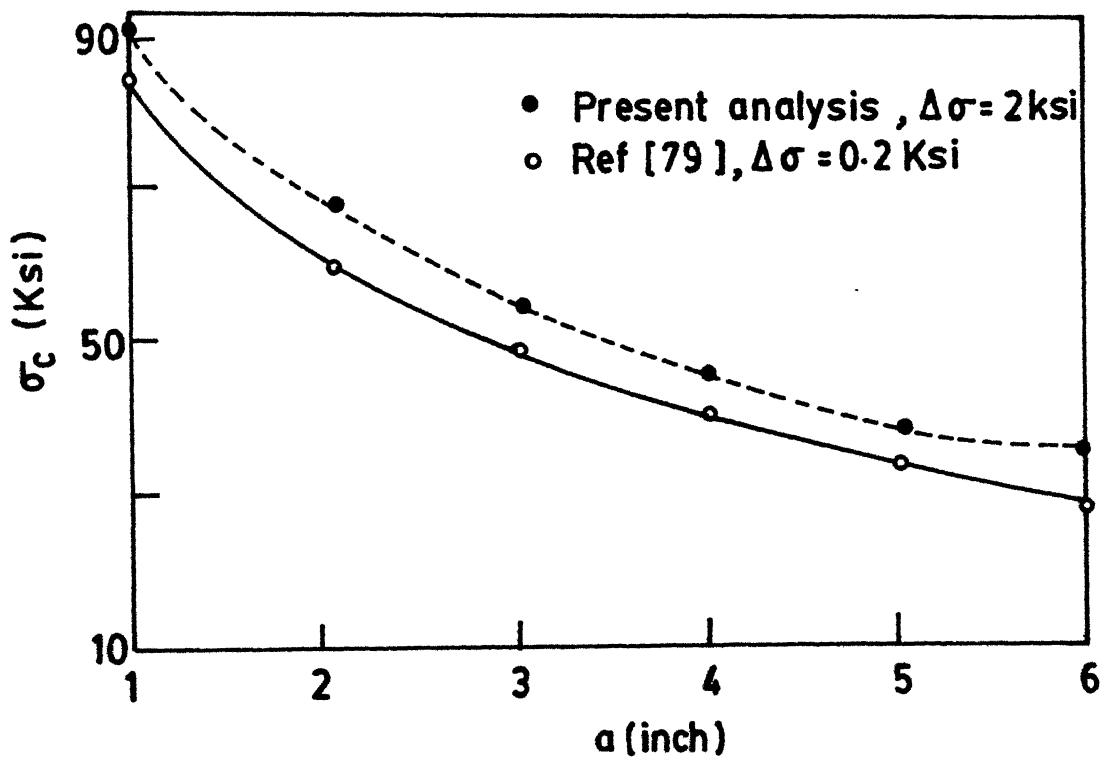


FIG. 5.9 Critical stress for global instability vs. initial half crack length

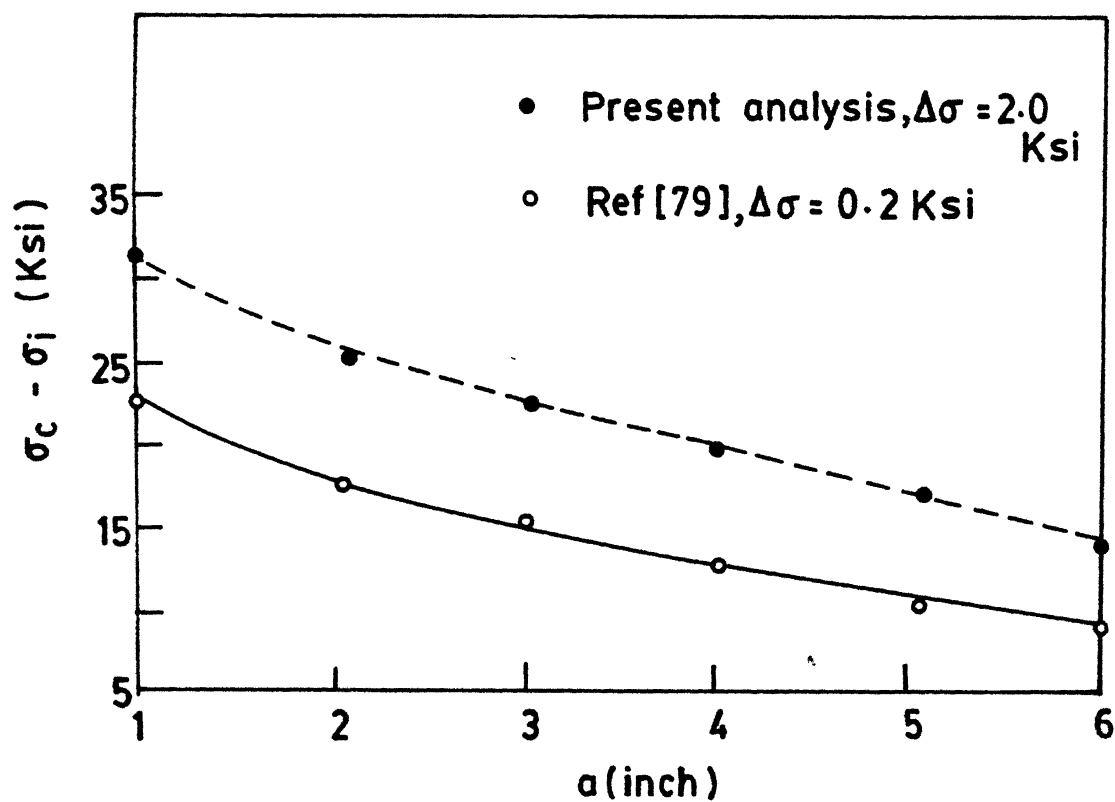


FIG. 5.10 Variation of difference of critical stress for instability and crack initiation,  $\sigma_c - \sigma_i$ , vs. initial half crack length

steps taken is  $\Delta\sigma = 0.2$  ksi and crack was propagated by 93 steps to arrive at the instability point whereas in the present analysis load steps  $\Delta\sigma = 2.0$  ksi and number of propagation steps 14 are taken. This shows the strong load step dependence of stable crack growth.

Figure 5.9 shows variation of stress  $\sigma_c$  versus the initial crack length  $a$ . It is observed that  $\sigma_c$  decreases as  $a$  increases. In order to obtain an estimate of the stress increment from crack initiation to unstable crack extension, variation of  $(\sigma_c - \sigma_i)$  versus the initial crack length  $a$  is presented in Fig. 5.10. It is observed that stress increment needed to bring the crack tip to onset of global instability from initiation of propagation decreases with initial crack length.

## 5.6 Propagation of an Angled Notch

The quasi-static crack propagation analysis of an angled notch problem shown in Fig. 5.11 is studied in this section. The present analysis results are compared with the same analysis using finite element method as well as experimental results of Ingraffea [77]. The initial mesh used and the boundary conditions considered for coupled BEM-FEM analysis is shown in Fig. 5.13. The material is Indiana (Salem) limestone the properties of which are indicated in Fig. 5.11.

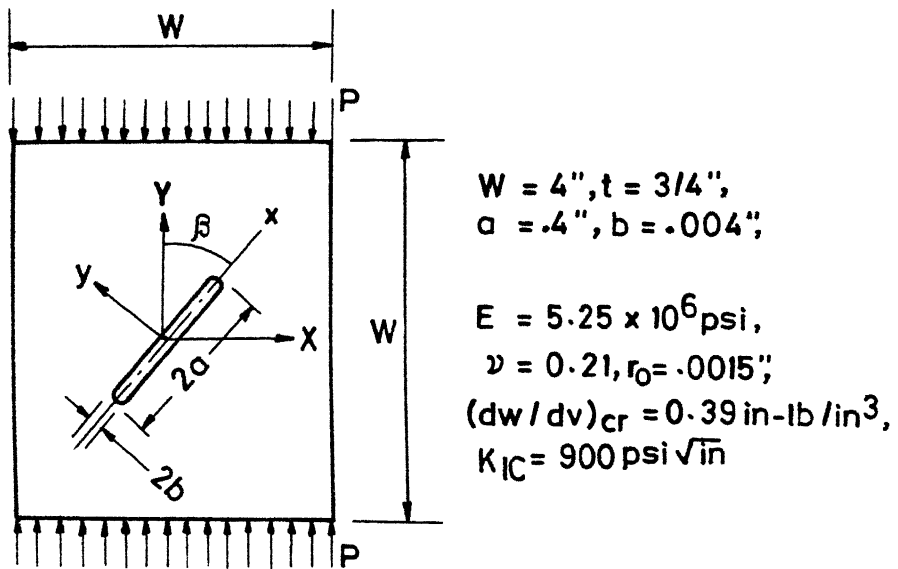


FIG.5-11 Angle notch plate, edge loaded in uniaxial compression

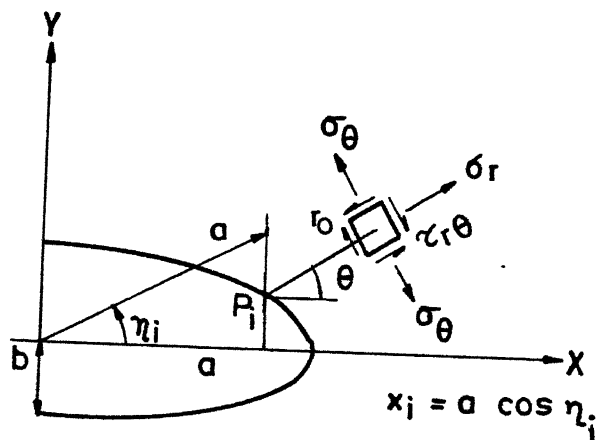


FIG. 5.12 Location of fracture initiation point on surface of ellipse

Since the problem is that of a notch, the surface layer energy concept [50] is used to predict the location for crack initiation point. The notch is assumed to behave like an idealized ellipse [78], with an effective semi-minor axis computed from,

$$b_{ef} = \sqrt{a\rho} \quad (5.1)$$

where  $a$  = notch half-length, 0.400 inch, and  $\rho$  = notch tip-radius, 0.004 inch. The point of maximum tensile tangential stress  $p_i$  is computed according to the positive root solution of

$$\tan \eta_i = \frac{b[a \sin^2 \beta - b \cos^2 \beta + \sqrt{a^2 \sin^2 \beta + b^2 \cos^2 \beta}]}{a(a+b) \sin \beta \cos \beta} \quad (5.2)$$

The geometric relationship between the eccentric angle  $\eta_i$  and  $p_i$  is depicted in Fig. 5.12. The values of x-coordinate of  $p_i$ ,  $x_i$ , and  $\eta_i$  for different notch angles are listed in Table 5.7.

Fracture initiation load,  $P_i$ , is calculated according to surface layer strain energy density theory for notch [50] which states that fracture initiation occurs when  $(dW/dV)_{\min}$  reaches a material critical value,  $(dW/dV)_{cr}$  at a distance  $r_0$  from the point  $p_i$  ( $x_i, \eta_i$ ) [as shown in Fig. 5.12], the fracture

TABLE 5.7 : Points of Fracture Initiation of an Angled Notch

$\beta$ (deg)	$\eta_i$ (deg)	$x_i$ (in)
0	0	0.4
15	5.5233	0.3981
30	8.153189	0.3960
45	11.9175	0.3914
60	18.4893	0.3794
75	34.5378	0.3295
90	90.0002	0.0



initiation point on the notch surface. The initiation load  $P_i$  versus notch angle  $\beta$  is then obtained by normalising this load according to,

$$P_i[\beta]/P_i[90^\circ] = F(\beta) \quad (5.3)$$

The results obtained by present analysis are compared with experimental values obtained by Ingraffea [82] in Fig. 5.14. As observed, the results agree quite well with the experimental data for  $\beta \leq 45^\circ$ . This is expected because for  $\beta > 45^\circ$ , notch is a poorer approximation to the ellipse for  $x_i$  predicted in this region. Also, as pointed out by Ingraffea [82], the area over which the tensile stress concentration acts increases rapidly with  $\beta$ . This implies that a slight local material weakness, such as grain boundary or weak grain, located any where within such a region may trigger fracture initiation well away from the theoretical point, resulting a wrong prediction of initiation load value.

The program FRAPCO is next used to perform crack trajectory prediction analysis. The notch angle  $\beta$  is taken to be  $45^\circ$ . The initial mesh used to perform the coupled BEM-FEM analysis is shown in Fig. 5.13. The load is gradually increased in steps and crack path is predicted from strain energy density theory, as outlined in Chapter 2. The computed crack path is compared with the experimentally obtained results and FEM results [77], in Fig. 5.15. The

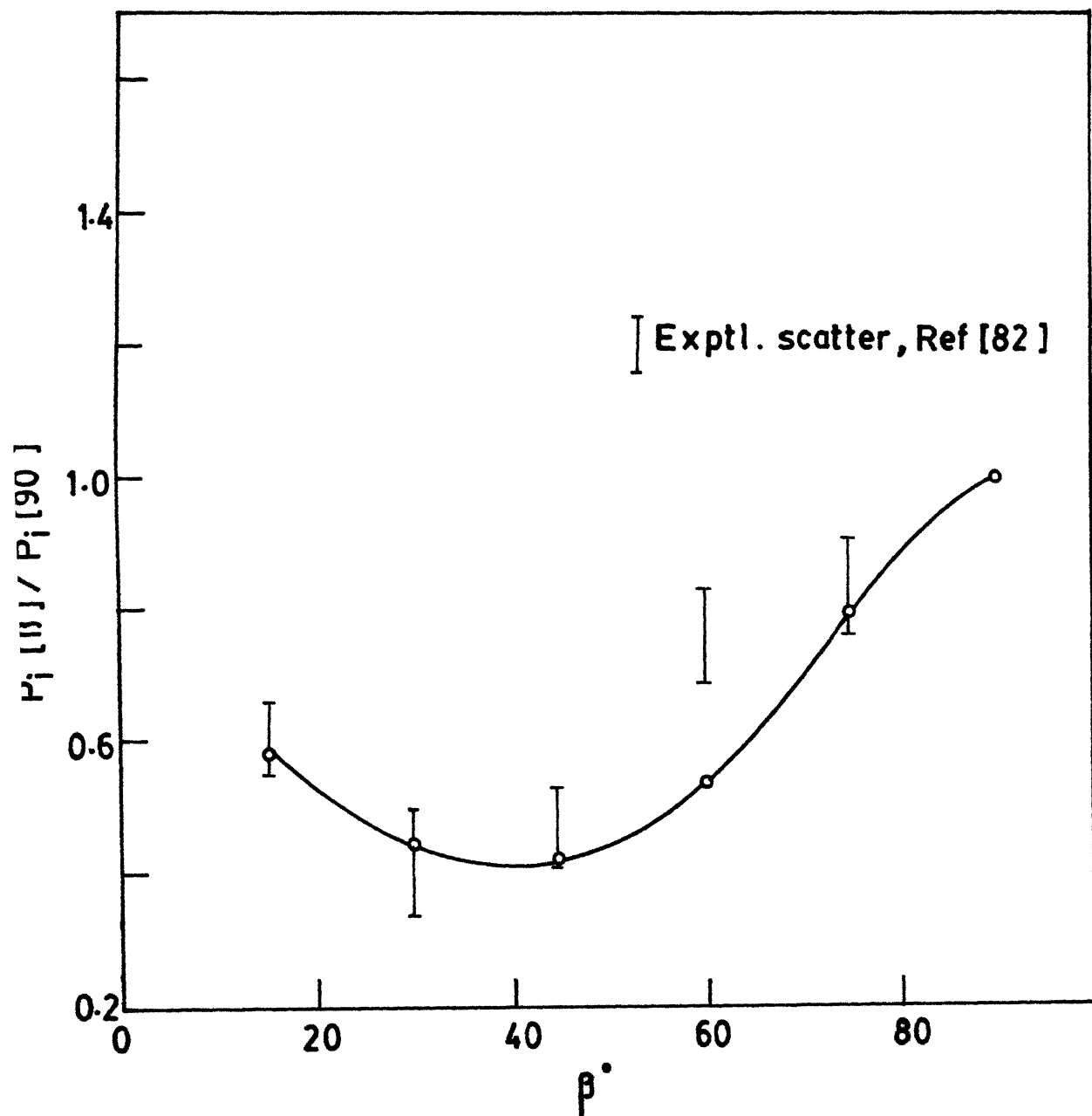


FIG. 5-14 Primary fracture initiation load

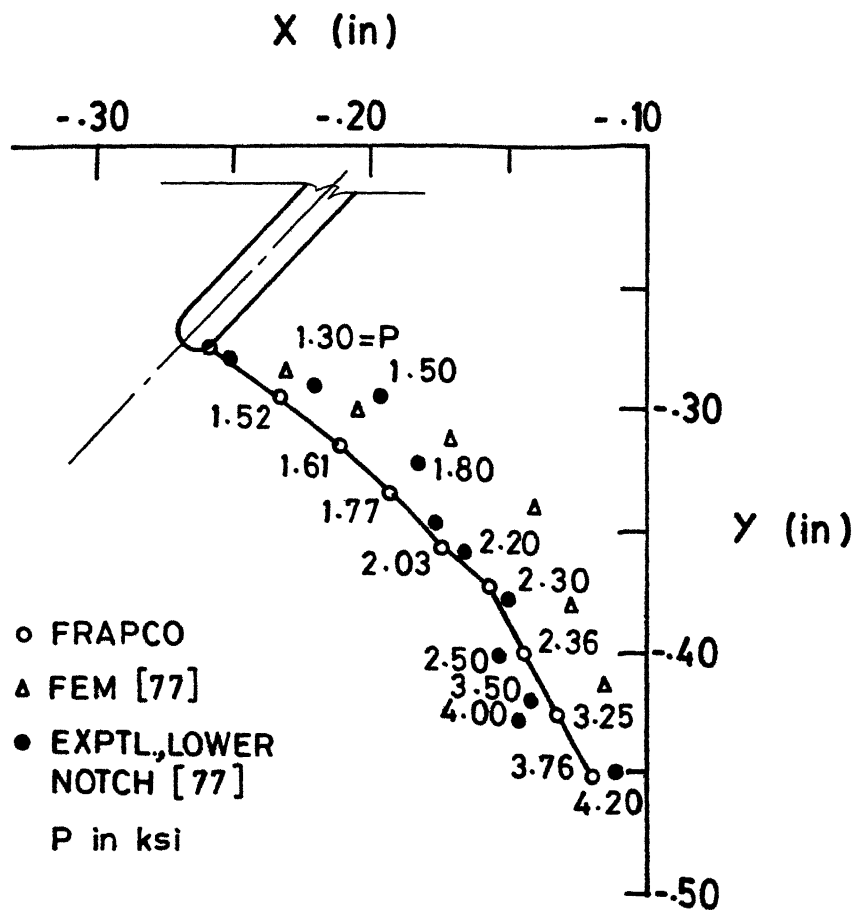


FIG. 5.15 Comparison of present analysis crack path with exptl. results and FE analysis

FEM results are obtained by maximum strain energy release rate theory. As seen from Fig. 5.15, there is good agreement between present analysis and experimental results. However, the finite element crack path prediction, particularly the load at certain steps, agree less with the present analysis. This discrepancy is most likely due to small errors in crack topology introduced by the irregularity of a finite element mesh. Using boundary elements around crack tip, the crack trajectory is modelled more smoothly in the present analysis and element distribution is also more regular. The resulting stress factor evaluation is thus more accurate, which in turn, results in a better load prediction at each crack step.

## CHAPTER 6

### CONCLUSIONS AND RECOMMENDATIONS

#### 6.1 Conclusions

The results of the present study have shown that an interactive discrete crack propagation analysis code based on a coupled BEM-FEM is a very efficient technique. The followings are the specific conclusions of the present work:

1. The coupling of BEM and FEM exploits the advantages of both methods. By using boundary elements around the crack tip, the following advantages are gained -  
(a) better accuracy of solution, (b) ease of discretizing a moving boundary, (c) ease of the use of singular elements, (d) considerable saving in computer time due to less number of nodes and elements as ~~only~~ boundary needs to be discretized. On the other hand, use of FEM away from crack tips offer the following advantages - (a) better efficiency of FEM for analysis of finite bodies with large dimensions, (b) use of locally based interpolation functions which produce narrowly banded final set of equations, (c) availability of frontal solution technique which requires less core storage and has got resolution facility.

2. Considerable saving in both time and storage requirements can be achieved by using a coupled BEM-FEM compared to FEM. This, combined with the fact that a unified approach can be adopted in finite element and boundary element regions' mesh generation, data base design, stiffness matrices' assembly etc. make such coupled BEM-FEM an attractive choice as it could be incorporated into an existing interactive finite analysis system with minor modifications.
3. Ability of a program to perform numerical analysis by BEM alone or FEM alone or by a coupled BEM-FEM is very advantageous as the user can use any one of these methods at his own discretion.
4. Transfinite mapping mesh generation technique could be used to generate mesh for finite element region and boundary element region equally efficiently.
5. The winged edge data structure is very efficient for interactive finite element and coupled boundary element - finite element analyses because - (a) All topological adjacency queries can be satisfied in constant or linear time, (b) data base can be modified to reflect topology change as a result of crack propagation without reorganizing unmodified portion of the data, (c) program response time grows at a rate no more than linearly with the problem size, (d) higher level operator could be defined which hide the inherent complexity of creating

and modifying this type of data structure.

6. Integrating all the aspects of a fracture mechanics analysis system such as fracture propagation theory, boundary and finite element methods, mesh generation, data base design etc. in a single code can aid in the technique and theory development cycle. Any new fracture propagation criterion and algorithm could be tested and accepted (or rejected) very efficiently and easily.

## 6.2 Recommendation for Further Study

The computer code developed herein, though quite versatile and general, many improvements over the present algorithms can be made. Also modular nature of the program leaves the possibility of incorporating new algorithms wide open. Few suggestions towards the development of a comprehensive fracture mechanics code by working on the present program are enlisted in the following lines.

- 1) The program FRAPCO relies heavily on virtual memory capability of the machine. A data structure, such as in [60] to take into account the restriction on the in-core storage, easy checking and access to each of the arrays involved and use of auxiliary memory could be employed to run the program in a medium size non-virtual memory machine.

- 2) In corporation of adaptive numerical integration scheme [ 60 ] in boundary element formulation is suggested.
- 3) Special crack tip elements [ 66 ] , [ 84 ] other than quarter point elements could be used to improve the accuracy of the results. Particularly attractive is the Hermitian cubic singular elements [ 66 ] as they do not require any division of problem domain into subdomains to model the crack surface. Thus multidomain BEM formulation, which requires additional effort from the user and also introduces error in the solution by constricting the solution along the interface to conform to the functional variation of the elements, can be avoided.
- 4) Extension to elastic-plastic analysis with few finite elements around crack tips and boundary elements away from it could be achieved very easily.



## REFERENCES

1. Engle Jr. R.M., "CRACKS, a FORTRAN IV Digital Computer Program for Crack Propagation Analysis", AFFDL-TR-70-107, October, 1970.
2. Chang J.B., Szamosi M., Liu K.W., "A User Manual for a Detailed Level Fatigue Crack Growth Analysis Computer Code, Volume 1 - The CRKGRO Program", AFWAL-TR-81-3904, 1981.
3. Chang J.B., Szamosi M., Liu K.W., "A User Manual for a Detailed Computer Program to Predict Fatigue Crack Growth on Flight-by-Flight Basis - The FLTGRO Program", AFWAL-TR-81-3904, 1981.
4. Walsh, P.F., "The Computation of SIF by a Special Finite Element Technique", Int. Jr. Solids Struct., 7, pp. 1333-1342, 1971.
5. Wilson W.K., "Some Crack Tip Finite Elements for Plane Elasticity" in : Fracture Toughness, ASTM-STP 514, 1971.
6. Tracy D.M. and Cook T.S., "Analysis of Power Type Singularities using Finite Elements", Int. Jr. Numer. Meth. Engg. 11, pp. 1225-1235, 1977.
7. Akin J.E., "The Generation of Elements with Singularities", Int. Jr. Numer. Meth. Engg., 8, pp. 537-545, 1974.
8. Tong, P., Pian T.H.H. and Larssy S., "A Hybrid Element Approach to Crack Problems in Plane Elasticity", Int. Jr. of Numer. Meth. Engg., 7, pp. 297-308, 1973.
9. Atluri S.N., Kobayashi A.S. and Nakagaki M., "An Assumed Hybrid Finite Element Model for Linear Elastic Fracture Mechanics", Int. Jr. Fract. 11, pp. 257-271, 1975.
10. Atluri S.N., Bass B.R., Bryson J.W. and Kthiresan K., "OR-FLAW : A Finite Element Program for Direct Evaluation of K-Factors for User Defined Flaws in Plates, Cylinders and Pressure Vessels Nozzel Corners", NUREG/CR-2494, ORNL/CSO/TM-165, April, 1982.
11. Saouma V.E. and Ingnaffea A.R., "Fracture Analysis of Discrete Cracking", Colloq. Adv. Mech. Reinf. Concrete, IABSE, Delft, The Netherlands, pp. 413-436, 1981.
12. Gerstle W.H., Martha L.F., Ingraffea A.R., "Finite and Boundary Element Modelling of Crack Propagation in Two and Three Dimensions", Engg. with Computers, 2, pp. 167-183, 1987.

13. Wawrzynek P.A., Ingraffea A.R., "Interactive Finite Element Analysis of Fracture Process - An Integrated Approach", Theoretical and Applied Fracture Mechanics, To appear.
14. Cruse, T.A., "The Surface Crack: Physical and Computational Solution", (Ed. : Swedlow J.L.), ASME, N.Y., pp. 153-170, 1972.
15. Cruse T.A. and Wilson R.B., "Advanced Applications of Boundary Integral Equation Method", Nucl. Engg. Design, 46, pp. 223-234, 1978.
16. Cruse T.A. and Synder M.D., "Boundary Integral Equation Analysis of Cracked Anisotropic Plates", Int. Jr. Fract., 11(2), pp. 315-328, 1975.
17. Kishimoto K., Yamaguchi I. and Tachihara, M., "Elastic Plastic Fracture Mechanics Analysis by Combination of Boundary and Finite Element Methods", Proc. 5th Intr. Conf. on Boundary Elements, Hiroshima, Japan, 1983.
18. Blandford G.E., Ingraffea A.R., Liggett J.A., "Two Dimensional Stress Intensity Factor Computations using the Boundary Element Method", Int. Jr. Numer. Meth. Engg., 17, pp. 387-404, 1981.
19. Perucchio R., Ingraffea A.R., "An Integrated Boundary Element Analysis System with Interactive Computer Graphics for 3-d Linear Elastic Fracture Mechanics", Comput. Struct., 20 (1-3), pp. 157-171, 1985.
20. Ingraffea A.R., Blandford G.E. and Liggett J.A., "Automated Modelling of Mixed Mode Fatigue and Quasi-static Crack Propagation using Boundary Element Method". ASTM STP-791, 1, 1984.
21. Zienkiewicz O.C., Kelly D.W. and Buttriss P., "Coupling of the Finite Element Method and Boundary Solution Procedures", Int. Jr. Numer. Meth. Engg., 11, pp.355-375, 1977.
22. Shaw, R.P., "Coupling Boundary Integral Method to Other Numerical Techniques" - Int. Symp. on Recent Adv. BEM, Southampton, England, July, 1978.
23. Brebbia, C.A. and Gergiou P., "Combination of BEM and FEM in Elastostatics", Appl. Math. Modelling, 3, 1979.
24. Kelly D.W., Mustoe G.G.W. and Zienkiewicz O.C., "Coupling of Boundary Element Methods and Other Numerical Methods", in: Developments in BEM-I (Ed.: Banerjee P.K.), pp. 251-285, Applied Sc. Pub. Ltd., London, 1979.

25. Tullberg G. and Batheus L., " A Critical Study of Different Boundary Element Stiffness Matrices", Proc. Fourth Int. Conf. on Boundary Element Methods in Engg., Southampton, England, September, 1982.
26. Hartman F., "The Derivation of Stiffness Matrices from Integral Equation" in: Boundary Element Methods (Ed.: Brebbia C.A.), Springer-Verlag, 1981.
27. Beer G., "Implementation of Combined Boundary Element-Finite Element Analysis with Application in Geomechanics" in : Developments in BEM-4 (Ed. Banerjee P.K. and Watson J.O.), Elsevier Applied Sc. Pub., London, 1986.
28. Cavendish J.C., "Automatic Triangulation of Arbitrary Planer Domains for the Finite Element Method", Int. Jr. Numer. Meth. Engg., 8, pp. 679-696, 1974.
29. Boubez, T.I. et.al., "Mesh Generation for Computational Analysis", Comput. Aided Engg. Jr., October, pp.190-201, 1986.
30. Lee Y.T., "Automatic Finite Element Generation Based on Constructive Solid Geometry"- Ph.D. Thesis, Mechanical Engineering Department, University of Leeds, Leeds, U.K., 1983.
31. Thacker W.C., Gonzalez, A. and Putland G.E., "A Method for Automating the Construction of Irregular Computational Grids for Storm Surge Forecast Models", Jr. of Computational Phys., 37, pp. 371-387, 1980.
32. Kela A., Perucchio R. and Voelcker H.B., "Towards Automatic Finite Element Analysis", Comput. Mech. Engg., 5, July, 1986.
33. Yerry M.A. and Shephard M.S., "A Modified Quadtree Approach to Finite Element Mesh Generation", IEEE Comput. Graph. & Appl., Feb, pp. 39-46, 1983.
34. Hermann L.R., "Laplacian-Isoparametric Grid Generation Scheme" Jr. of the Engg. Mechanics Division, ASCE, 102, No EM5, pp. 749-756, Oct. 1970.
35. Lorensen W., "Grid Generation Tools for the Finite Element Analyst", First Chautanqua on Finite Element Modelling, (Ed: Conaway J.H.), Schaffer Analysis Inc., Mt. Vernon, N.H., pp. 119-136, 1980.

36. Buell W.R. and Bush B.A., "Mesh Generation - A Survey", Jr. Engg. Ind., Transaction ASME, 95(1), 1973.
37. Hall C.A., "Transfinite Interpolation and Application to Engineering Problems", in: Theory of Approximation, (Ed: Law and Sahney), Academic Press, pp. 308-331, 1976.
38. Haber R. et.al., "A General Two-dimensional Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mappings", Int. Jr. Numer. Meth. Engg., 17, pp. 1015-1044, 1981.
39. Gordon W.J., "An Operator Calculus for Surface and Volume Modelling" IEEE Comput. Graph and Appl., pp.18-22, Oct. 1983.
40. Thacker W.C., "A Brief Review of Techniques for Generating Irregular Computational Grids", Int. Jr. Numer. Meth. Engg., 15, pp.1335-1341, 1980.
41. Baumgart B.G., "A Polyhedron Representation for Computer Vision", AFIPS Proc., 44, 589-596, 1975.
42. Braid I.C., Hillyard R.C. and Stroud I.A., "Stepwise Construction of Polyhedra", in: Geometric Modelling", in: Mathematical Methods in Computer Graphics and Design, (Ed. : Brodile K.W.), pp. 121-143, Academic Press, London, 1980.
43. Yamaguchi F. and Tokieda T., "A Solid Modeller with a 4x4 Determinant Processor", IEEE Comput. Graphics and Appl., 5(4), 1985.
44. Weiler K., "Edge-base Data Structure for Solid Modelling in Curved Surface Environment", IEEE Comput. Graph. Appl., 5(1), pp. 21-40, 1985.
45. Woo T.C., "A Combinatorial Analysis of Boundary Data Structure Schemata", IEEE Comput. Graph. Appl., 5(3), pp. 19-27, 1985.
46. Mantyla M. and Sulonen R., "GWB : A Solid Modeller with Euler Operators", IEEE Comput. Graph. Appl., 2(9), pp. 17-31, 1982.
47. Wawrzynek P.A. and Ingraffea A.R., "An Edge Based Data Structure for Two Dimensional Finite Element Analysis", Engg. with Computers, 3, pp. 13-20, 1987.
48. Gdoutos E.E., "Problems of Mixed-Mode Crack Propagation", Martins Nijhoff Pub., The Hague, 1984.
49. Sih G.C., "A Special Theory of Crack Propagation Methods of Analysis and Solution of Crack Problems",

- in: Mechanics of Fracture-I (Ed: Sih G.C.), Noordhoff Intr. Pub., Leyden, pp. 21-45, 1973.
50. Sih G.C., "Surface Layer Energy and Strain Energy Density for a Blunted Crack or Notch", in: Prospects of Fracture Mechanics, (Ed: Sih G.C. et.al.), Noordhoff Int. Pub., Leyden, pp. 85-102, 1974.
  51. Williams, M.L., "On Stress Distribution at the Base of a Stationary Crack", Jr. Appl. Mech., 24, pp.109-114, 1957.
  52. Irwin G.R., "Analysis of Stress and Strain near the end of a Crack Traversing a Plate", Jr. Appl. Mech., 24, pp. 361-364, 1957.
  53. Sih G.C. and Liebowitz H., "Mathematical Theories of Brittle Fracture", in: Fracture, An Advanced Treatise, (Ed.: Liebowitz H.), 2, pp. 67-190, Academic Press, N.Y., 1968.
  54. Zienkiewicz O.C., "The Finite Element Method", 3rd Edition, McGraw Hill, London, 1977.
  55. Irons B.M., "A Frontal Solution Program for Finite Element Analysis", Int. Jr. Numer. Meth. Engg., 2, pp.5-32, 1970.
  56. Love, A.H., "A Treatise on the Mathematical Theory of Elasticity", Dover, N.Y., 1944.
  57. Brebbia C.A., Telles J.C.F., Wrobel L.C., "Boundary Element Techniques", Springer-Verlag, 1984.
  58. Stroud A.H. and Scerest D., "Gaussian Quadrature Formulas", Prentice-Hall, New York, 1966.
  59. Schren, E., "Computer Implementation of Finite Element Procedure", in: Numerical and Computer Methods in Structural Mechanics (Ed: Fenves S.J. et. al.), Academic Press, pp. 79-121, 1973.
  60. Doblaré M., "Computational Aspects of Boundary Element Method", in: Topics in Boundary Element Research' (Ed. : Brebbia C.A.), Springer-Verlag, N.Y., 1987.
  61. Cruse, T.A., "Two-dimensional BIE Fracture Mechanics Analysis", Appl. Math. Modelling, 2, pp. 287-293, 1978.
  62. Barsoum R.S., "On Use of Isoparametric Finite Elements in Linear Fracture Mechanics", Int. Jr. Numer. Meth. Engg., 10, pp. 25-37, 1976 .
  63. Ingraffea A.R. and Manu C., "Stress Intensity Factor Computation in Three Dimensions with Quarter Point Crack Tip Elements", Int. Jr. Numer. Meth. Engg., 12(6), pp. 235-248, 1978.

64. Volait F., "Three-dimensional Superelements by Direct Boundary Integral Equation Method for Elastostatics", Proc. Third Int. Seminar on Boundary Element Methods, Irvine, California, July, 1981.
65. Patterson C. and Sheikh M.A., "Interelement Continuity in the Boundary Element Method", in: Topics in Boundary Element Research-1 (Ed: Brebbia C.A.), pp. 123-140, Springer-Verlag, 1984.
66. Watson J.O., "Hermitian Cubic Singular Elements for Plane Strain", in: Developments in BEM-4 (Ed: Banerjee P.K. and Watson J.O.), Chapter 1, pp. 1-28, Elsevier Appl. Sc. Pub., London, 1986.
67. Ho-Le K., "Finite Element Mesh Generation Methods: A Review and Classification", Computer Aided Design, 20(1), Jan./Feb. 1988.
68. Gordon W.J. and Hall C.A., "Construction of Curvilinear Coordinate Systems and Application to Mesh Generation", Int. Jr. Numer. Meth. Engg., 7(4), pp. 461-477, 1973.
69. Gordon W.J., "Blending Function Methods of Bivariate and Multivariate Interpolation and Approximation", SIAM Numer. Anal., 8, pp. 158-177, 1971.
70. Mantyla M., "A Note on the Modelling Space of Euler Operators", Computer Vision, Graphics and Image Processing", 26, pp. 45-60, 1984.
71. Ingraffea A.R., "Fracture Propagation in Rock", in: Mechanics of Geomaterials (Ed. : Bažant Z.) John Wiley and Sons Ltd., 1985.
72. Woo T.C. and Wolter J.D., "A Constant Expected Time, Linear Storage Data Structure for Representing 3D Objects", IEEE Trans. Systems. Man and Cybernatics, SMC-14(3), pp. 510-515, May/June, 1984.
73. Baer A., Eastman C. and Henrion M., "Geometric Modelling: A Survey", Computer Aided Design, 2(5), pp. 253-272, Sept., 1979.
74. Zienkiewicz O.C. and Phillips D.V., "An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Coordinates", Int. Jr. Numer. Meth. Engg., 3, pp. 519-528, 1971.
75. Tada H., Paris P.C. and Irwin G.R., "The Stress Analysis of Cracks, Handbook", Del Research Corpn., Hellertown, Pennsylvania, 1973.

76. Bowie, O.L., "Rectangular Tensile Sheet with Symmetric Edge Cracks", Jr. of Appl. Mech., 31, pp. 208-212, 1964.
77. Ingraffea A.R. and Heuze F.E., "Finite Element Models for Rock Fracture Mechanics", Int. Jr. Numer. Analytical Meth. Geomech., 4, pp. 25-43, 1980.
78. Inglis, G.R., "Stresses in a Plate due to Presence of Cracks and Sharp Corners", Trans. Roy Inst. Naval Architects, 60, pp. 219-230, 1913.
79. Gdoutos E.E., "Stable Growth of a Central Crack", Theoretical and Appl. Fract. Mech., 1, 1984, pp.139-144.
80. Kitagawa H. and Yuuki, R., "Analysis of Branched Cracks under Biaxial Stresses", in: Advances in Research on the Strength and Fracture of Materials, 3A - Analysis and Mechanics (Ed. : Taplin D.M.R.), pp. 201-211, Pergamon Press, 1978.
81. Bowie O.L., "Analysis of an Infinite Plate Containing Radial Cracks Originating from the Boundary of an Internal Circular Crack", Jr. Math. Phys., 35, pp. 60-71, 1956.
82. Ingraffea A.R. and Ko H.Y., "Determination of Fracture Parameters for Rock", Proceedings of Ist USA-Greece Symp. Mixed-mode Crack Propagation, Athens, Greece, pp.349-365. Aug., 1980.
83. Isida M., "Effect of Width and Length on Stress Intensity Factors of Internally Cracked Plate under Various Boundary Conditions", Int. Jr. Fract. Mech., 7, pp. 301-316, 1971.
84. Tanaka, M., "A New Family of Crack Elements for Stress Intensity Factor Computation in Elastostatics by Boundary Element Method", in: Boundary Elements VIII (Ed. : Brebbia C.A.), Springer-Verlag, 1986.

105914

TH  
620.1126 Date 105914

M312C This book is to be returned on the  
date last stamped.

.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

ME-1888-M1-MAN-CRA